Analyse van discrete-tijd-wachtlijnsystemen met meerdimensionale toestandsruimte

Analysis of Discrete-Time Queueing Systems with Multidimensional State Space

Stijn De Vuyst

Promotoren: prof. dr. ir. H. Bruneel, prof. dr. ir. S. Wittevrongel
Proefschrift ingediend tot het behalen van de graad van
Doctor in de Ingenieurswetenschappen

Vakgroep Telecommunicatie en Informatieverwerking
Voorzitter: prof. dr. ir. H. Bruneel
Faculteit Ingenieurswetenschappen
Academiejaar 2005 - 2006

UNIVERSITEIT
GENT

ISBN 90-8578-038-1
NUR 959
Wettelijk depot: D/2005/10.500/38

Aan de grote en de kleine tovenaar

— *nil desperandum* —

Voorwoord

Deze scriptie geeft een overzicht van het onderzoekswerk dat ik de laatste jaren heb verricht op het gebied van de performantie-analyse van wachtlijnsystemen. Eenvoudig gezegd bestudeert men in deze discipline van de ingenieurswetenschappen het gedrag van systemen waarin entiteiten op de een of andere manier moeten *wachten*. Sinds het begin van de 20ste eeuw, met de opkomst van de telefonie, vormt de telecommunicatiewereld een steeds belangrijkere drijfveer voor het doen van dergelijk onderzoek. Wanneer men een hedendaags (digitaal) computer- of telecommunicatienetwerk bekijkt, zijn daarin ontelbare plaatsen te onderscheiden waar pakketjes informatie tijdens hun reis doorheen het systeem tijdelijk moeten wachten alvorens zij verder kunnen verwerkt of doorgezonden worden. Zeker nu de transmissie van informatie steeds sneller en in grotere hoeveelheden verloopt, is het van belang om inzicht te verkrijgen hoe een concreet systeem zich zal gedragen onder een bepaalde werklast en hoe men eventueel het ontwerp van deze systemen kan verbeteren.

De werking van realistische communicatiesystemen is meestal gebaseerd op enkele eenvoudige *deterministische regels*. De machine volgt een bepaald protocol, vastgelegd in zijn software of hardware. Ondanks de eenvoud van de regels is het toch niet voor de hand liggend om te voorspellen hoe het systeem zich zal gedragen eens het in werking is. De oorzaak is dat de *aansturing* of *invoer* meestal op een grillige, onzekere manier gebeurt, wat tot gevolg heeft dat het systeem zelf eveneens op een onzekere, schijnbaar toevallige manier evolueert. Echter, we kunnen de onzekerheid aan de invoer vatten in een abstract *stochastisch model*, waarbij aan elk exemplaar van de invoer dat mogelijks kan optreden een bepaalde *kans* wordt verbonden. Bij één bepaald exemplaar van de invoer reageert het systeem zonder onzekerheid volgens zijn protocol of programma zodat het resulterende systeemgedrag vastligt. Door nu de kansverdeling over alle mogelijke invoersexemplaren in rekening te brengen, zijn we in staat de kans te berekenen waarmee het systeem een bepaald gedrag zal vertonen, wat nuttige voorspellingen inzake de performantie mogelijk maakt.

In het meest ideale geval bestaat de opzet van de wachtlijntheorie als toegepaste discipline er dus in de volgende dialectische cyclus te vervolmaken:

(a) de werking van een reëel systeem te vatten in een vereenvoudigd, stochastisch model,
(b) vervolgens dit model te gaan doorrekenen,
(c) de bekomen resultaten te gebruiken om uitspraken te doen over de performantie van het reële systeem en
(d) (eventueel) de bekomen inzichten gebruiken om betere systemen te ontwerpen.

Het is duidelijk dat men hierin des te beter zal slagen wanneer het model dichter bij het werkelijke systeem aanleunt. Hoe beter het model, hoe kleiner de discrepantie tussen het voorspeld en het realistisch gedrag van een systeem. De keerzijde van de medaille is echter dat het toevoegen van allerlei finesses, nuances en complicaties die men ontwaart in het systeem ertoe kunnen leiden dat het model – weliswaar heel compleet en accuraat – te log en te gecompliceerd wordt om het nog wiskundig te kunnen ‘oplossen’.

Het werk gepresenteerd in deze scriptie wil juist hierin een bijdrage leveren. Het doel is een conceptueel inzicht te verwerven in enkele specifieke fenomenen die van belang zijn in communicatiesystemen. Deze fenomenen werden in de ‘traditionele’ modellen verwaarloosd, niettegenstaande zij een belangrijke impact kunnen hebben op het gedrag van het systeem, zoals zal blijken. We stellen ons hier de vraag: hoe kan men deze modellen realistischer maken? Hoe kan men ze uitbreiden om ze beter te laten beantwoorden aan de werkelijke omstandigheden? In Hfst. 2 bijvoorbeeld wordt het feit dat bronnen informatiestromen genereren op een *gecorreleerde* manier in rekening gebracht, waar eerdere modellen veronderstellen dat dit op een ongecorreleerde manier gebeurt. Evenzo wordt in Hfst. 3 een verbetering voorgesteld die rekening houdt met het feit dat fouten bij het verzenden van informatie eveneens vaak op een gecorreleerde manier voorkomen. In beide gevallen gaan we zowel kwalitatief als kwantitatief na wat het effect is van deze complicaties. Echter, de prijs voor deze betere modellen wordt betaald in de vorm van een complexere rekenopgave. Hfst. 4 tenslotte is in deze optiek wat afwijkend, omdat wij daar een geheel nieuwe wachtlijndiscipline voorstellen en volledig analyseren.

Ik moet toegeven dat men als academisch onderzoeker er vaak meer om bekommerd is de wiskundige kant van de zaak (puntje (b), het doorrekenen van het model) tot een goed einde te brengen, dan de resultaten daadwerkelijk te gaan toepassen in een realistisch systeem. Dat aspect wordt vaak aan anderen overgelaten. Een belangrijke reden daartoe vormt het aanvoelen dat een mooi model qua toepasbaarheid niet noodzakelijk beperkt is tot dat ene protocol of communicatie-apparaat. Vaak kunnen op zijn minst de *kwalitatieve* resultaten van een model inzicht verschaffen in een veel bredere klasse van systemen. Meer nog, hetzelfde kan gezegd worden over de wiskundige methodes die gebruikt worden om van het model te komen tot praktisch berekenbare resultaten. Niettegenstaande de rekenopgave complexer is, slagen wij er toch in om deze in alle volledigheid en op een exacte manier door te voeren. Kortom, de *analyse* van het model vormt een op zichzelf staand stuk werk met zijn eigen toegevoegde waarde. Zo komt het ook dat men op den duur enkel nog het abstracte model voor ogen heeft en dat men met het begrip *systeem* eigenlijk steeds het *gemodelleerde* systeem bedoelt. In tegenstelling tot de aard en het toepassingsgebied van de behandelde modellen is de *methode* van analyse wel degelijk uniform doorheen dit werk. Steeds wordt de evolutie van het (gemodelleerde) systeem opgevat als een Markovketting met een meerdimensionale toestandsruimte. Door gebruik te maken van meerdimensionale genererende functies kunnen we de kansverdeling in evenwicht berekenen van de systeemtoestand, wat dan weer de aanzet is tot het verkrijgen van praktische performantiematen zoals de systeembezetting en de vertragingstijden. Deze benaderingswijze is ook wel gekend als de *methode van de toegevoegde variabele* en is samen met het gebruik van genererende functies zowat het handelsmerk van het onderzoek bij SMACS.

Het is aangewezen hier mijn dank uit te spreken tegenover een aantal personen die op uiteenlopende manieren hebben bijgedragen tot deze scriptie. Vooreerst wil ik mijn begeleiders en promotoren Sabine Wittevrongel en Herwig Bruneel bedanken omdat zij mij de gelegenheid hebben geboden dit soort boeiend onderzoek te verrichten. De modellen die in deze scriptie behandeld worden, zijn veelal geïnspireerd op hun eerder werk, evenals de methodes voor het analyseren ervan. Het moet ook vermeld dat de sfeer in de SMACS-groep er altijd een is geweest waarin het onderzoeksgebeuren in alle vrijheid kan verlopen, wat m.i. essentieel is voor de kwaliteit van het werk, zeker op lange termijn. Dank gaat ook uit naar de andere leden van SMACS, met wie het steeds prettig samenwerken is. In het bijzonder vermeld ik Dieter Fiems, Joris Walraevens en Bart Vinck, collega's en vrienden met wie ik de werkplek deel. Dieter en Joris hebben ook stukken van de tekst nagelezen en door hun gewaardeerde opmerkingen er essentieel aan bijgedragen. Verder bedank ik ook alle andere collega's van de vakgroep TELIN, waaronder ook Annette Nevejans voor haar inzet betreffende de sociale cohesie in de groep. Mijn ouders Roland en Ivette bij wie ik altijd terecht kan, vrienden en familie bedank ik voor hun steun en vertrouwen. Tot slot bedank ik ook mijn lieve Geneviève, voor het geduld dat zij met mij weet op te brengen. Na een onproductieve periode in mijn werk heeft zij me liefdevol geholpen terug een beetje discipline aan de dag te leggen en meer doelgericht te werken, dit nu reeds twee jaar lang.

Stijn, september 2005.

Samenvatting

In deze scriptie behandelen we drie verschillende wachtlijnmodellen met betrekking tot enkele typische fenomenen die optreden in moderne digitale communicatienetwerken, hetzij vaste, hetzij draadloze. In deze systemen wordt informatie gemanipuleerd in de vorm van logische eenheden van vaste lengte die we *pakketten* noemen. Van de wachtlijnen wordt een stochastisch model opgesteld in discrete tijd. We onderstellen met name dat de tijd wordt opgedeeld in *slots* en dat veranderingen in het systeem slechts kunnen plaatsvinden op de slotgrenzen. Wij stellen een exacte analyse voor die uitvoerig gebruik maakt van *probabiliteitsgenererende functies* (pgf's). Elk van de drie modellen kan gezien worden als een *uitbreiding* van meer klassieke modellen, in de zin dat de gemaakte onderstellingen specifiek werden aangepast om beter te beantwoorden aan de realistische omstandigheden waarin de wachtlijn wordt gebruikt. Naast de concrete resultaten die we behalen met betrekking tot de wachtlijnperformantie in deze modellen, bestaat de bijdrage van dit werk er ook in om de kracht en de veelzijdigheid van de gebruikte analytische technieken te demonstreren.

- Met het eerste model bestuderen we de gelijktijdige impact op de wachtlijnperformantie van twee verschillende soorten correlatie aanwezig in het aankomstproces. De wachtlijn stelt in dit geval een *statistische multiplexer* voor die berichten bundelt afkomstig van een grote groep gebruikers. De *berichten* bestaan hierbij uit een algemeen verdeeld aantal pakketten. De pakketstroom die de multiplexer krijgt te verwerken is dan onderhevig aan volgende correlaties. Ten eerste is er een *primaire correlatie* als gevolg van het feit dat de berichten 'aankomen als een trein', d.w.z. aan een snelheid van één pakket per slot. Ten tweede gebeurt het genereren van de berichten zelf ook niet onafhankelijk van slot tot slot, wat resulteert in een *secundaire correlatie*. Er wordt namelijk ondersteld dat het aantal nieuwe berichten per slot door de gebruikers gegenereerd, wordt gemoduleerd door een Markovproces met twee toestanden. Eerst bepalen we een impliciete relatie voor de gezamenlijke evenwichtsverdeling van de systeemvariabelen, waaruit we uitdrukkingen afleiden voor de momenten en de staartverdeling van het aantal pakketten in de wachtlijn. Daarna concentreren we ons ook op de vertragingstijd die de berichten ervaren in het systeem. Daarbij onderstellen we dat de pakketten behandeld worden volgens een 'First-In First-Out' (FIFO) discipline en dat pakketten die gelijktijdig aankomen in willekeurige volgorde worden opgeslaan. We verkrijgen de gemiddelde waarde voor zowel de vertragingstijd als de verwerkingstijd van een willekeurig bericht en geven nauwe grenzen aan voor hun staartverdeling.
- Het tweede model dient voor de analyse van het Stop-and-Wait ARQ protocol *(Automatic Repeat reQuest)* of SW-ARQ, dat tot doel heeft een betrouwbare transmissie van pakketten over een inherent foutgevoelig kanaal te verzekeren. Het aantal pakketten dat transmissie aanvraagt, wordt onafhankelijk ondersteld van slot tot slot. Deze pakketten worden opgeslagen in een *zenderbuffer* terwijl ze hun transmissie afwachten over het kanaal naar de ontvanger toe. Wanneer een pakket de ontvanger bereikt, wordt gedetecteerd of er een transmissiefout is opgetreden en de zender wordt met een bevestigingsbericht op de hoogte gebracht van de al dan niet foutieve ontvangst. Bij Stop-and-Wait ARQ stuurt de zender slechts een enkel pakket en wacht dan op diens bevestiging. Bij een negatieve bevestiging wordt

het betreffende pakket opnieuw verstuurd, maar bij een positieve bevestiging mag het de wachtlijn verlaten en wordt begonnen met de transmissie van een nieuw pakket. Het belangrijke aspect van dit model is onze onderstelling dat de fouten in het kanaal niet onafhankelijk optreden van elkaar, maar gemoduleerd worden door een Markovproces met twee toestanden, net als de gebruikersomgeving in het vorige model. Het onderstellen van dergelijke correlatie in het foutproces is van speciaal belang bij communicatie over een *draadloos* kanaal, waarbij de fouten typisch gegroepeerd optreden, in zgn. 'bursts'. Voor dit model bepalen we eerst de evenwichtsverdeling van de systeemtoestand en van het aantal pakketten in de zenderbuffer. Daarna bekijken we de maximale doorvoer van het systeem en de verdeling van de pakket-vertragingstijd. Aansluitend leiden we ook een boven- en ondergrens af voor de kans dat de wachtlijn leeg is.

- Het derde model behandelt een wachtlijn die onderscheid maakt tussen twee types van pakketten en de verwerking regelt volgens een nieuw soort wachtlijndiscipline die we de *Reservatiediscipline* noemen. Het doel is hier om de vertraging van pakketten met hoge prioriteit (type 1) te reduceren ten koste van pakketten met lage prioriteit (type 2). De gebruikelijke manier om dit te realiseren is om Absolute Prioriteit (AP) te verlenen aan pakketten van type 1. Echter, met AP is het verschil in vertraging soms drastischer dan beoogd en kan dit verschil niet worden aangepast. Onze oplossing bestaat erin om N *reservatieplaatsen* te voorzien in de wachtlijn voor toekomstige aankomsten van type 1 pakketten. Bij aankomst van een type 1 pakket neemt dit de verst gevorderde reservatie in en maakt een nieuwe reservatie aan het einde van de wachtlijn. De type 2 pakketten echter, nemen steeds plaats aan het einde van de lijn zoals bij FIFO. Op deze manier wordt de pakket-volgorde per type behouden en kan het verschil in vertragingstijd aangepast worden a.d.h.v. de parameter N. Eerst wordt het model met *één* reservatie behandeld, waarvoor de pgf, de momenten en de staartverdeling van de vertragingstijden van beide types pakketten worden verkregen. Daarna wordt de analyse uitgebreid voor $N > 1$ reservaties. De resultaten worden vergeleken met zowel de FIFO als de AP discipline.

Men zou kunnen opperen dat deze drie modellen zich respectievelijk concentreren op één van de drie basiscomponenten van een 'wachtlijnsysteem': het multiplexermodel bekijkt complicaties in het *aankomstproces*, het model van de SW-ARQ zender heeft een niet-triviaal *bedieningsproces* en de Reservatiewachtlijn heeft een speciale *wachtlijndiscipline*. De structuur van deze scriptie is als volgt. In het eerste hoofdstuk motiveren we de modellen vanuit de toepassingen in digitale communicatienetwerken. We gaan in op stochastische modellering in het algemeen en discrete-tijd-wachtlijnen in het bijzonder, de benodigde wiskundige middelen zoals discrete toevalsveranderlijken, pgfs en Markovkettingen. De algemene methodologie waarmee de wachtlijnanalyse wordt uitgevoerd en die we de *Discrete Supplementary Variable Technique* (DSVT) noemen, wordt eveneens kort geschetst. In respectievelijk de Hfstn. 2, 3 en 4 behandelen we elk van de bovenstaande modellen. Deze hoofdstukken kunnen volledig onafhankelijk van elkaar gelezen worden. De resultaten in elk hoofdstuk worden ruimschoots geïllustreerd met een aantal numerieke voorbeelden. Een appendix in Hfst. 3 geeft een overzicht van de literatuur rond ARQ protocollen en hun performantie-evaluatie. Daarbij gaan we specifiek in op de vraag in hoeverre het gebruikte kanaalmodel toepasbaar is voor draadloze communicatie.

Summary

We present and analyse three different queueing models that are motivated from phenomena occurring in modern communication networks, both wired and wireless. In these systems, information is handled in the form of logical fixed-size units called *packets*. The models are treated analytically in a stochastic discrete-time setting, i.e. assuming that time is divided in fixed *slots* and that changes in the system can only take place at slot boundaries. Also, the queues all have infinite capacity. The analysis we propose is exact at the packet level, and is characterised by an extensive use of *probability generating functions* (pgfs). Each of the presented models can be seen as an *extension* of more traditional models, in the sense that they have an additional layer of 'complexity'. Compared to these earlier models, we have refined the assumptions in order to capture the working conditions of the queue in a more realistic way. Besides the tangible results we obtain for the queueing performance in each of the three models, one of the main goals of this work is to demonstrate the power and the versatility of the analytic technique we use.

- With the first model, we study the simultaneous impact of two different types of correlation in the packet arrival process on the performance of a queue. The queue in this case, represents a *statistical multiplexer* to which messages arrive generated by an unbounded population of users. Each *message* in this context consists of a generally distributed number of packets. We assume the packet arrival process to exhibit the following two types of correlation. First, the messages arrive to the multiplexer as a 'train', i.e. at the rate of one packet per slot. This results in what we call the *primary correlation* in the arrival stream. Also, on a higher level, the arrival process contains an additional *secondary correlation*, resulting from the fact that the behaviour of the user population is governed by a two-state Markovian environment. Specifically, the state of the environment during a particular slot determines the number of new messages in that slot. We first derive an implicit relation for the joint distribution of the system state during equilibrium, from which expressions are inferred for the moments and the tail distribution of the number of packets in the queue. Using these results, we then also concentrate on the message delay distribution, under the important assumption of a First-In First-Out (FIFO) queueing discipline for packets, whereby packets that arrive during the same slot are stored in random order. We obtain the mean value of both the total delay and the transmission time of an arbitrary message. Additionally, we provide a reasonably tight upper and lower bound for the tail probabilities of the message delay.
- The second model deals with the analysis of the Stop-and-Wait ARQ (Automatic Repeat reQuest) protocol for the reliable transmission of packets over an error-prone channel. The number of packets per slot that request transmission is assumed to be independent and identically distributed (iid). These packets are temporarily stored in the *transmitter queue* to await their transmission over the channel towards the receiver. After a packet reaches the receiver, it is checked for errors and the transmitter is notified by returning a feedback message over the backward channel. Under Stop-and-Wait ARQ (SW-ARQ), the transmitter sends a single packet and then waits for the feedback message. If the packet was in error, it is simply transmitted over the channel again. On the other hand, if the transmission was suc-

cessful, the packet in question is removed from the queue and the transmission of another packet starts. The important feature of this model is the fact that the packet errors in the channel are not iid, but modulated by a two-state Markov chain, just like the user environment in the previous model. This assumption is particularly important when the *wireless* channel is considered, where errors typically occur in a bursty, correlated manner. We first derive the equilibrium distribution of the system's state and the queue content, after which we focus on the throughput and the distribution of the packet delay. For the latter, we use the spectral decomposition theorem from linear algebra. Additionally, lower and upper bounds for the probability of the queue being empty are also given.

- The third model is concerned with a special kind of queueing discipline, which we term the *Reservation discipline*. The objective of this scheduling mechanism is to reduce the delay perceived by packets that are delay-sensitive (type 1), at the cost of allowing higher delays for the best-effort packets (type 2). The 'traditional' way of providing such differentiation in delay is to give Absolute Priority (AP) to the packets of type 1. With AP however, the differentiation may be larger than desired and cannot be controlled. Our solution is to introduce N *reserved places* in the queue for future arrivals of type 1. Whenever a packet of type 1 enters the queue, it takes the position of the most advanced reservation and creates a new reservation at the end of the queue. Type 2 arrivals on the other hand, always take place at the end of the queue in the usual FIFO manner. This way, per-type FIFO is attained, as well as a stochastic delay difference between the types that can be controlled by the parameter N. We first concentrate on the Reservation queue with only *one* reserved place. Explicit results are obtained for the pgf, the moments and the tail distribution of the delay of either traffic type. Then we extend the analysis to the case of $N > 1$ reservations and give algorithmic solutions to obtain the same results. The results are compared with those of the FIFO and AP disciplines.

One could say that these three models each focus on one of the three components of a 'queueing system': the multiplexer model is concerned with the *arrival process* (i.e. the input), the SW-ARQ transmitter queue has a non-trivial *service process* (the output), while the Reservation queue has a special *queueing discipline*.

The structure of this thesis is as follows. In the first chapter we introduce the reader to digital communication networks and their ubiquitous need for buffering. We explain the morale behind stochastic modelling in general and the use of discrete-time queueing models as a means of predicting buffering behaviour. Then the necessary mathematical tools, such as discrete random variables, probability generating functions (pgfs) and Markov chains are reviewed. Finally, by means of two short examples, we illustrate the general methodology used to analyse each of the above queueing models, which we call the *Discrete Supplementary Variable Technique* (DSVT). The chapters 2, 3 and 4 can be read independently of each other and each treat one of the models in detail. The results are discussed and illustrated with ample numerical examples. An appendix of chapter 3 gives some more background information on ARQ methods and their performance modelling. We specifically focus on the question in how far the channel model we used in that chapter is applicable for wireless communication.

Contents

Chapter 1

Introduction

1.1 Digital computer networks and queueing

In general, the purpose of a communication system is to manipulate *information* and to convey it from one place to another. The way in which this happens depends on the nature of the information, but also on the available technology. Until the 19th century, the technology to communicate over longer distances was limited to physically transporting something or someone carrying the information, like a letter or a messenger. Things changed when in the course of that century scientists started to gain better understanding of the electromagnetic force, which lead to technological developments like the telegraph by S. Morse (1830), the telephone by A. Bell (1876) and the wireless telegraph by G. Marconi (1895). In these devices, information like a text or a sound is converted into an analog electrical signal and subsequently sent over an electric wave conductor like a wire or into free space with an antenna. The main advantage of this technology now is that the medium *itself* no longer has to be transported, as it already extends from the sender to the receiver. Instead of transporting matter, information can now be conveyed as pure energy in the form of an electromagnetic wave signal travelling at nearly the speed of light. In the course of the 20th century, people learned to use the electromagnetic medium with increasing efficiency by developing better wave conductors, antennas, modulation techniques and coding schemes. The latter techniques are used to put as much information as possible into the transmitted signal, without compromising the chance of correctly retrieving this information at the receiver too much. Most contemporary communication networks like television cable networks, computer networks, fixed and mobile telephone networks are still based on the use of the electromagnetic medium, be it wired or wireless. A fairly recent technology is to convert information to a sequence of light-pulses and to transport them over an *optical* wave conductor. Although light is electromagnetic in nature as well, this is an essentially new technology which allows to transport information in even larger quantities and at greater speeds. At this moment, the optical transport of information over glass fiber cables is common practice, but the technological possibilities to *process* that information are still very limited. Therefore, in practical optical systems, essential parts of the data have to be converted very often from optical to electrical form and back.

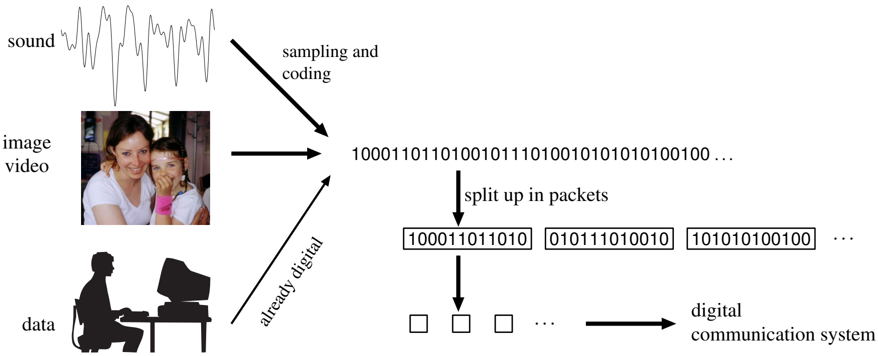


Figure 1.1: Analog information like sound or images can be converted into digital form. When the bit sequences, or messages, have to be transported over a digital communication network, they are split up into packets conform to a format that can be handled by the network.

1.1.1 Digital communication

In the meantime, an important paradigm shift had occurred as to the *representation* of the information that is used in communication systems. Whereas in early systems, information processing was a matter of manipulating *analog* signals, the current systems work with information that is represented *digitally*. The shift from analog to digital information processing was powered by the availability of a new technology as well. In 1947, by relying on new insights in solid-state physics, Bardeen and Brattain succeeded in building the first semiconductor transistor. When later, these components were integrated on small silicon chips, this sounded the beginning of the modern computer age. Computers, so we know, are capable of processing huge amounts of information but do require this information to be presented in digital form, i.e. as a sequence of *bits*. A bit is the smallest possible unit of information with value either 0 or 1. Auditive and visual information for us people, can only be interpreted as an analog signal. However, by using a sufficient number of bits this signal can be represented with arbitrary precision in digital form, as illustrated in Fig. 1.1. By using efficient coding techniques, often especially adapted to a specific type of information, the number of required bits can sometimes be kept surprisingly small. There are many advantages to the use of a digital representation as opposed to an analog one. In principle, bit sequences are more resilient to degradation and noise, being the reason why a computer can make millions of computations per second without ever making one error. Also, digital processing is conceptually simpler, more versatile and often less costly than processing signals with analog electronics.

The way in which we represent and process information is determined by the 'best' available technology. New technologies in turn, emerge from discoveries in physics. Charles Babbage's difference engine (1822) manipulated numbers in decimal representation and relied on simple mechanics to do so[1]. Nowadays, computers process

[1] Ironically, Babbage never managed to actually make a prototype *because* of technological problems.

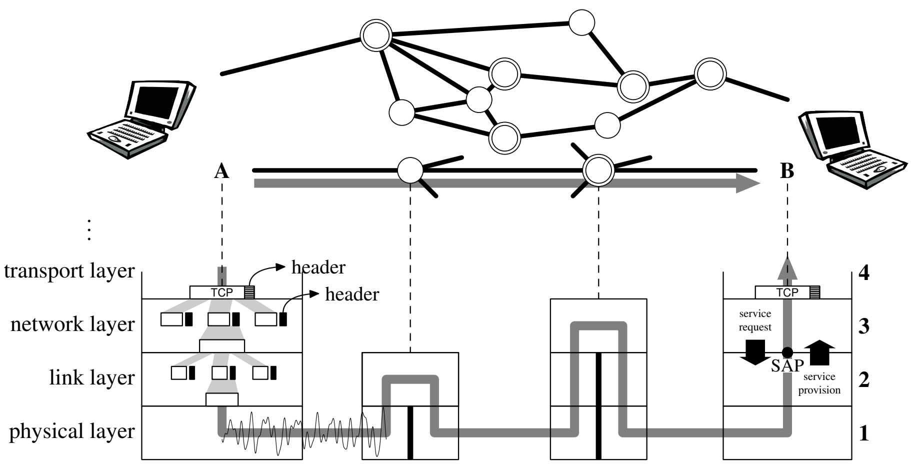


Figure 1.2: Layered structure of a communication network according to the OSI reference model. Every layer is responsible for a number of functions which are performed by its own packet-based protocol.

numbers in binary format using semiconductor electronics. In the future, we may represent information in yet another way if the quantummechanical properties of atoms can be harnessed into a feasible computing technology. In quantum computing [155], information is represented by *qubits* rather than bits, taking values among the different quantum states of a photon or atom. Conceptually, quantum computing promises to solve (some) complex problems in a much faster way than can now be done. However, like with Babbage in his days, technology still falls behind.

Considering the success of the digital concept, it is only logical that nowadays communication systems process information in the form of bit sequences as well. One of the many advantages is the fact that different kinds of information with heterogeneous origins, can be treated uniformly and can be easily combined. Whether a certain bit sequence represents an image, a text or a sound sample, the system does not always have to make specific arrangements, because it is all bits and bytes, regardless of their origin or interpretation. In this thesis, we do not deal with the *coding* aspect of digital communication, but rather focus on the *networking* aspect, or the dynamics of moving pieces of information about in a communication system. Specifically, one such dynamic phenomenon of particular importance is *queueing*.

1.1.2 Layered network architecture

For transporting information over long distances, digital systems indeed depend on a networking infrastructure, a digital communication network. On the most basic level, a network consists of geographically separated *nodes* that are interconnected with a number of *links* over which communication can effectively take place. When a source presents information to one of the nodes, the purpose of the network is to transport it to a destination at another node as good and as fast as possible. We will discuss further what is meant here by 'good'. The practical accommodation of many simultaneous and heterogeneous traffic streams between a multitude of sources and destinations dispersed over the different nodes is a very complex task. To keep the implementation of a communication network tractable and manageable, such network is most often designed according to a hierarchical *layered* architecture, as conceptually depicted in Fig. 1.2. In such an architecture, each of the separate layers is responsible for an exclusive part of the functions that the network as a whole has to fulfill. As such, the scope of a layer extends horizontally over different nodes and vertically over some particular functions. Every layer takes decisions independently of the other layers, according to a predefined set of rules called a *protocol*. A specific designation of protocols to a number of adjacent layers is a protocol *stack*. Also, the same layers at two different nodes are often called *peer layers* to indicate the spatial separation and the fact that both are at the same level of hierarchy.

The practical implementation of the protocol used in a certain layer is self-contained and does not depend on the implementation of the protocols in the other layers. This ensures a certain interoperability between network devices from different vendors. Obviously, the layers do pass information to each other, but communication is only allowed between *adjacent* layers, i.e. the layer directly above or the one below. Most protocols are packet-based, which means that *inside* a layer the data is handled in the form of indivisible groups of either fixed or variable size. Depending on the specific protocol in which they are used, these units are termed *datagrams*, *cells*, *blocks* or *frames*. Typically, the lower level protocols employ units of small size with fixed size in which case we consistently use the generic term *packets*. In upper layers, there is often no predefined upper limit for the size of the information units which we then call *messages*.

Let us assume that, as in Fig. 1.2, the protocol on layer 4 in node A wants to send a message to the peer layer in node B. Layer 4 in A cannot send the message to B directly, as there is only a physical link between the nodes at the lowest layer and layer 4 can only communicate with layers 3 and 5. Rather, the message is conveyed indirectly, by descending the hierarchy in node A to the lowest layer, being physically transported to node B and there ascending the hierarchy until the peer layer is reached. Specifically, layer 4 in node A issues a request for service to layer 3 asking to deliver the message to layer 4 in node B. It is now the further responsibility of layer 3 to provide this service in such a way that all the functions resorting under layer 3 are taken care of. To this end, the original message is split up in packets conforming to the format required by the layer 3 protocol and each of these are passed on to layer 2. After further reformatting of the information into layer 2 packets, these are passed to layer 1 in fulfillment of the layer 2 functions. The main function of layer 1 then, is to provide the physical transport to node B. Once there, the information climbs up the hierarchy again and at each layer, the original packets are reconstructed from the packets of the layer below. Eventually, layer 3 will deliver the necessary packets for layer 4 in B to reconstruct the original message. On the way from A to B, among other things it must be decided which further route is to be taken. As routing is typically a function of layer 2 or 3, the information must climb up and descend a few layers in some intermediate nodes as well.

The communication with adjacent layers, i.e. either the *service requests* to the layer below or the *service provisioning* to the layer above, is conducted over a so-called *service access point* (SAP) which is a common interface between the two protocols. In our example above, when layer 4 wants to send a message from A to B, it does so in the form of a service request to layer 3 via the SAP in between. The way in which layer 3 carries out this task is of no further interest to layer 4 until the moment layer 3 delivers the message to layer 4 in B, thereby providing the requested service. From the viewpoint of layer 4 then, it is as if there is only one intermediate communication partner between A and B, the protocol of layer 3. Note that every layer, except for the highest and lowest one, in the hierarchy is both *service user* for the layer below and *service provider* for the layer above. Also, keep in mind that there can be multiple instances of 'users' in a layer that simultaneously rely on the services of the layer below in a parallel way.

Although a layer in the source node can only pass information to the adjacent layers in the same node, it is foremost its intention to communicate with the peer layer in the destination node. But how does layer 3 in B for example, *know* what to do with the packets it receives? The specific instructions in layer 3 from A to B are stored in an extra block of information called a *header* and appended to the actual data or the *payload*. The header contains all the control information that the protocol needs to perform its functions with regard to the payload. Together, payload and header constitute a valid packet conform to the format specifically required by the layer 3 protocol. Note that the distinction between payload data and header is meaningful *only* to the current layer and a very complex situation arises once the packets are passed to lower layers. As shown in Fig. 1.2 the whole message, including the header, at layer 4 is split up into the payloads of layer 3 packets. To each of these, a specific layer 3 header is appended, resulting in valid layer 3 packets. The same happens when the layer 3 packets are passed to layer 2. So in fact, as the information progresses to lower layers, more and more *overhead* information is added in the form of headers.

The OSI (Open System Interconnection) reference model [161, 221] dating from the early 80s distinguishes a total of seven layers, to be divided in two groups. The first group consists of the *application layer* (7), the *presentation layer* (6) and the *session layer* (5) which are involved with application-specific issues and generally implemented only in software. The application layer is the closest to the user and the *application* it is running, like a web browser, a database or an online game. The practical protocols at layer 7 are therefore also the most familiar ones to the average computer user. Examples are: surfing the web via HTTP (HyperText Transfer Protocol), e-mail via SMTP (Small Message Transfer Protocol) and POP3 (Post Office Protocol), file transfer with FTP (File Transfer Protocol) and using a remote computer with Telnet. The functions of this layer are to check the availability of intended communication partners, the identification and authorisation of users, the security and synchronisation of distributed applications and generally, to execute the user's commands at the highest level. As such, the implementation of these protocols is often very specific to the kind of user application. The presentation layer deals with the *format* in which the data will be exchanged. If necessary, the data is translated to a platform-independent representation as to avoid compatibility problems. The functions of the presentation layer are: data conversion, coding, encryption, compression and negotiation with regard to the best data format. The session layer on level 5 takes care of establishing, maintaining and breaking up a temporary connection (a session) between applications.

Contrary to the upper layers, the lower layers of the OSI model are not application-oriented but network-oriented. They provide an application-independent system for the reliable *transport* of data over the network. Some of the necessary functions, like error detection and correction, are performed on more than one layer but always within the scope of the layer's responsibility. The functions are assigned to the four layers as follows:

- The *transport layer* (4) oversees the end-to-end transmission from the location of a source user to the location of the destination user. The principal function is to provide a reliable and sequential packet delivery by using error recovery mechanisms. Another function is *flow control*, i.e. to adjust the flow of data from one device to another to ensure that the receiving end can handle all of the incoming data. This way, the protocol tries to avoid congestion further up in the network. The most well-known layer 4 protocol is TCP (Transmission Control Protocol) which implements both flow and error control as opposed to UDP (User Datagram Protocol) which implements neither. UDP is therefore much simpler and only provides an unguaranteed service to applications for which reliability is less important.
- The *network layer* (3) is mainly concerned with logical addressing and routing, i.e. selecting a path between two end systems that may be located on geographically diverse subnetworks. A subnetwork or *segment* in this context is essentially a single network infrastructure often using the same implementation solutions in all the nodes. The subnetworks are connected to each other with gateway routers and choosing a suitable route through the subnetworks is a layer 3 responsibility. For instance, IP (Internet Protocol) is the protocol that connects many diverse subnetworks all over the world into one large network we know as the Internet.
- The *link layer* (2) regulates the reliable transmission of data *within* a subnetwork. It provides physical routing, using a flat address space that is confined to the subnetwork only. In fact, choosing a physical path through the subnetwork at the level of the link layer is called *switching*, as opposed to the term *routing* which is used at the level of the network layer. The link layer may also provide error and flow control, but only for the direct transmission over one physical link.
- The *physical layer* (1) is the only layer that is capable of actually using the network's physical medium to convey a bit stream to another node. In both wired and wireless networks, such characteristics as voltage levels, data rates, maximum transmission distances, physical connectors and so on are defined by the physical layer's specification. Many different technologies exist for implementing the physical layer, such as using coax cables, twisted-pair cables or the wireless medium.

Not all communication protocols that are in use today adhere strictly to the OSI model. In fact, there are few who do. Most protocols like IP and TCP can be situated more or less at a certain level but may take care of some functions of the adjacent layers as well. Ethernet (IEEE 802.3) or ATM (Asynchronous Transfer Mode) are technologies that largely combine the functionalities of both layer 1 and 2. Some protocols on the other hand control only part of a layer's functions. The functions of e.g. the link layer are often split up into a MAC layer (Media Access Control) and an LLC layer (Logical Link Control). After the OSI reference model was presented, specific protocol standards were developed that implement each layer's functions. However, these new protocols never gained much popularity because another de facto standard based on combining TCP and IP (TCP/IP) was already being widely deployed. As discussed in e.g. [184] both the OSI model and the TCP/IP have serious disadvantages. The OSI model today is no longer of practical importance, but still provides a very useful theoretic and conceptual reference for discussing communication networks. The converse is true for TCP/IP that is used on a large scale, but does not arise from a theoretically well-founded model.

1.1.3 Quality of service

An increasingly important notion in digital communication networks is *Quality of Service* (QoS). By QoS, we essentially mean a measure indicating how well the demands and needs of the network users[2] are met. Providing adequate QoS to the traffic streams is what we have indicated earlier as one of the two main duties of a communication network: provide transport as fast as possible and as *good* as possible. Obviously, different kinds of applications have different requirements with regard to the transport of the packets they generate. In the table below taken from [184], some common applications are listed with their sensitivity to four *service criteria*.

Application	*Reliability*	*Delay*	*Delay jitter*	*Bandwidth*
email	high	low	low	low
file transfer	high	low	low	average
web browsing	high	average	low	average
remote login	high	average	average	low
audio streaming	low	low	high	average
video streaming	low	low	high	high
telephone	low	high	high	low
videoconferencing	low	high	high	high

For the first four applications, reliability is of particular importance. They need to be absolutely sure that their packets are delivered correctly, without errors. On the other hand, it is often the case that such applications are much less sensitive to delay and jitter (i.e. to the variability in the delay of subsequent packets). For instance, when transferring a large file over the network, one cannot tolerate that parts of the file get lost or are corrupted during transport because that would make the complete file useless to the end user. However, as long as it is reliable, the user does not mind waiting a few seconds more for the transfer to complete. The converse is true for the other four applications in the table: multimedia applications or so-called *real-time* applications. If the packets do not reach the end user within a certain time, they can no longer be used to reconstruct the audio or video signal and become useless. For e.g. voice telephony the delay should be kept as low as possible. A difference of only 200ms is already noticeable and a conversation delay of more than 800ms is irritating and confusing [144]. Reliability on the other hand, is much less of an issue here,

[2] Recall that in this thesis the term 'user' can mean *any* source that offers a flow of packets to be processed or transmitted through the network. A user is therefore not always a physical person, but can be an application or a protocol in some layer requesting service to the next protocol in the stack.

since multimedia is usually coded in such a way that an acceptable quality can still be reproduced even if some parts of the packet stream are missing or erroneous. Notice however that the 'delay' requirement for the streaming applications is low as opposed to telephony and videoconferencing. The reason is that streaming traffic is one-way only, and a *constant* delay does not affect the decoding at the receiver (although jitter does).

Ideally, all users strive for maximum QoS, i.e. no errors, no packet loss, minimal delay, no jitter and maximum bandwidth. As the resources of the network are limited however, a compromise has to be made, preferably on the QoS criteria that the traffic is least sensitive to. For our purposes, it suffices to distinguish between only two classes of traffic, based on the sensitivity to the first two service parameters in the table above:

❶ *Real-time traffic*: delay-sensitive, but loss-tolerant
The packets in this kind of data flow foremost need to be on their destination as soon as possible with small jitter. Bad reliability, i.e. the fact that a few packets get lost or corrupted now and then, is not a problem if kept within limits.

❷ *Non-real-time traffic*: loss-sensitive, but delay-tolerant
The applications generating this type of traffic cannot afford a packet to be lost, but generally do not mind if these packets take more time to reach the destination.

As a convention in this thesis, we refer to these classes as type 1 and type 2 respectively. This classification is very rudimentary, but suffices for the discussion in this thesis. In reality, many more QoS levels and dimensions are being identified, with less coarse granularity.

The first packet-based networks and protocols were designed with only very limited QoS provisioning in mind. All data packets were largely the same to the network, resulting in a 'best-effort' performance for all types of traffic without guarantees. However, as networks grew larger and were used to transport more types of heterogeneous traffic, the need for QoS provisioning was recognised and new networking standards were developed that can effectively increase the QoS experienced by specific classes of users. In fact, one of the most active fields of research in telecommunications today is undoubtedly the development of new protocols and devices that can increase the level of QoS in large networks transporting a lot of heterogeneous traffic. Making a network QoS-aware involves special arrangements as to provide better service (with regard to a specific service criterium) to one or more *selected* flows. Evidently, increasing the QoS for one flow often implies that the other flows experience worse QoS. Nevertheless, this is precisely the kind of behaviour that was aimed for and allows to distribute the available network resources to the different flows in such a way that their specific needs are taken into account. Many different approaches are used to increase the QoS in a network, both at the level of the user (such as e.g. traffic shaping) and at the level of the network (congestion avoidance, special scheduling mechanisms, etc.), see e.g. [96].

The models we discuss in chapters 3 and 4 are specifically related to the challenge of increasing QoS in the network. The subject of chapter 3 is ARQ, a technique used at the link layer to increase the reliability with repeated retransmissions. As we will see, increasing reliability comes at the cost of increased delay, so mainly non-real-time traffic (type 2) will benefit from this. In chapter 4, we discuss an entirely new scheduling mechanism that can statistically lower the perceived delay of type 1 traffic at the cost of the type 2 delay.

1.1.4 The need for queues

The models discussed in this thesis are inspired by important dynamic phenomena that occur in layered communication networks. As information traverses through the different layers of the network (both horizontally and vertically), it is handled by many different protocols and is constantly being reorganised and reformatted. In this process, there are numerous occasions throughout the network in which a packet of information has to *wait* before it can proceed further towards its destination. No matter how large the processing power available, communication devices are still sequential machines and can handle only one or at most a few parallel tasks at a time. Hence, it is inevitable that at certain points in the protocol stack more packets need a certain service than can momentarily be provided. In order to function properly, a modern packet-based communication network therefore requires *queues* or *buffers*, i.e. facilities to *store* packets for a while and retrieve them at a later time when the resources are available to process them.

Let us consider a specific example that demonstrates how networks designed for higher efficiency and speed have an increasing need for queues. In the context of layered networking, it often occurs that multiple traffic streams have to be sent over the same link, as shown in Fig. 1.3, an operation which is called *multiplexing*. After the packets reach the other end of the link, they are *demultiplexed* again into the original separate data streams. At least in concept, this situation can arise at any layer, whenever multiple 'users' or sources want to use a single network resource. The link in this case has a limited capacity, which means that no more than a certain amount of packets can be transported per time unit. In Fig. 1.3, the capacity of the link would be fully used if 4 sources were to send packets continuously.

There are two ways to divide the capacity of the link between the different users. Communication systems of the early days, like e.g. the Public Switched Telephone Network (PSTN) use a technique called *circuit-switching* whereby part of the link capacity is exclusively reserved to each of the incoming streams. Once a user is granted such a connection, it can send packets over its designated portion of the link at the assigned rate. In Fig. 1.3 **(a)**, the four users each have one quarter of the link's capacity which means that no room is left to accommodate packets from *other* users. The advantage of this method is that whenever a user has a packet to send, it can do so directly without delay. Moreover, this impeccable service is always guaranteed, even if the user sends packets at maximum rate all of the time. However, as in the example of Fig. 1.3, most users are not active continuously but often exhibit extended periods of silence. Under these circumstances, it is clear that circuit-switching is inefficient: packets from other users are denied access to the link even if some of the admitted users do not use their assigned capacity.

The alternative to circuit-switching is *packet-switching*, shown in Fig. 1.3 **(b)**, which allows the link to be used more efficiently. With this method, no user can make an exclusive claim to any of the link's capacity. Instead, the packets of *all* users are first aggregated and seen by the system as coming from one collective user. The packets

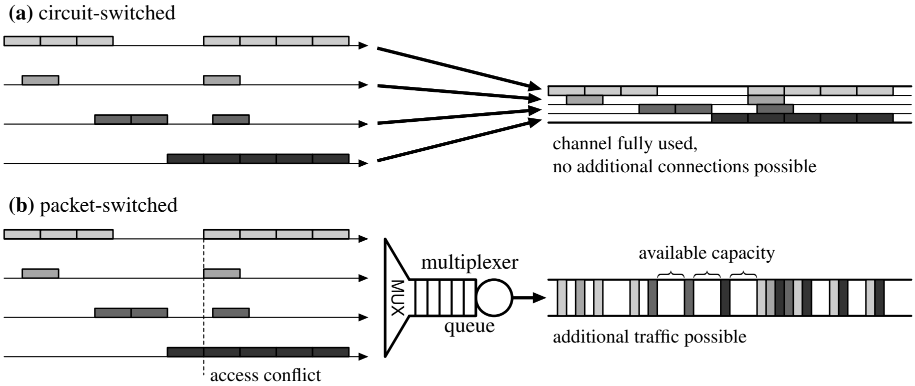


Figure 1.3: Two ways to send multiple traffic streams over the same link: circuit-switching in **(a)** and packet-switching in **(b)**. In the latter case, a queue needs to be implemented to avoid access conflicts.

in the aggregated stream are then transmitted at the native speed of the link. The main advantage is that there is no predefined upper bound to the number of users that can use the link simultaneously. As long as this limit is not reached, the link has available capacity for additional packets. As such, the increased efficiency obtained with packet-switching mainly stems from the interrupted nature of the users' traffic patterns, which is a statistical property. The gain in efficiency is therefore sometimes referred to as 'statistical gain'. This gain comes at the price of increased implementation cost however. As in Fig. 1.3 **(b)**, it is possible that two or more users send a packet at the same instant, or at a time when the link is occupied transmitting another packet. Such *access conflicts* have to be resolved by using a buffer where packets can temporarily be stored. In most modern computer networks, packet-switching is preferred over circuit-switching.

A system specifically designed to aggregate multiple traffic streams into a single output stream of packets is called a *multiplexer* (MUX), which must always have some buffering capabilities in order to realise a statistical gain. Unlike circuit-switching however, statistical multiplexing *does* have consequences for the QoS perceived by the users. Specifically, the fact that packets have to queue up before accessing the link introduces an additional delay in their delivery. Also, if the available space in the queue is finite (as it always is in practice), there is a chance that the queue becomes fully occupied and that additional packets get lost anyway. As such, one could say that the requirement of maximum QoS to a *few* users in the case of circuit-switching is relaxed in favour of a diminished guarantee in QoS to *many* users in the case of packet-switching. The way in which the QoS is affected depends on the queue size[3]: large queues are good for loss-sensitive traffic while small queues are good for delay-sensitive traffic.

[3] And on other aspects such as the queueing discipline as well, see further.

1.1.5 Queueing theory

Evidently, queues do not only occur in packet-based networks or telecommunication devices in general, but are important in many other engineering disciplines as well, like production planning, operations research, information technology, software engineering etc. Even in everyday life, we are confronted with queues everywhere: people queueing up to buy tickets at a movie theater, being stuck in a traffic jam or in a line at the counter of a grocery store. Sometimes the behaviour of such queues has a profound impact on our lives, like e.g. being on a waiting list for an organ transplant or other medical treatment.

In view of their broad importance, it is not surprising that the study of queues has become a scientific discipline in its own right. *Queueing theory* is concerned with the mathematical description and analysis of queueing phenomena. Given some quantitative assumptions about the structure, the dynamic operation of the queue(s) and its environment, it is the goal of the theory to make accurate predictions about the behaviour of the queue. With respect to digital communication again, predicted measures are e.g. the amount of packets present in the queue (*queue content*) or the amount of time spent in the queue by a particular packet (*delay*). Such mathematical results prove to be useful in practice, as they can assist the design and dimensioning of the buffering facilities in communication devices.

The publication that spawned queueing theory was [78], dating from 1909. It was by the hand of A.K. Erlang who is generally considered to be the pioneer in this area. In [38], an overview of his most important work can be found which is mainly concerned with the measurement and analysis of telephone traffic and the dimensioning of switching centers. For a detailed historical overview of queueing theory, we refer to [35, 177]. Classic reference works reviewing methods of queueing analysis include [59, 105, 114] and more recently also [34, 64].

1.2 Dealing with uncertainty, the stochastic approach

As a dynamical system, a queue is usually influenced by certain processes in the environment in which it operates. In most cases, these processes are too complex to be described deterministically, i.e. with an exact account of all causes and effects that lead to the exhibited behaviour. If we consider for example the statistical multiplexer of Sec. 1.1.4 again, it is clear that the number of packets in its queue is influenced by the number of users, as well as by how many packets they generate and how these packets are spread over time. Given the exact time stamps of all packets, it would be a matter of simple calculation to conclude on the number of packets in the queue at any given time. Gathering such exact knowledge could for instance be done by logging the traffic trace at the input of the multiplexer, but in that case, one might as well measure the queue content too. Nevertheless, describing the packet generation with complete certainty *in advance* is not always possible as this would require exact knowledge of all underlying causes and their specific interactions that eventually lead to a packet being generated or not. Moreover, such deterministic description is not even desirable, because the obtained conclusions would be applicable in that specific situation (of a particular input traffic trace) only. As an alternative, describing a process *stochas-*

tically offers a shift in trade-off. A single stochastic description fits *many* different traces and the conclusions concerning the queueing behaviour are valid for all these traces. However, the price of this broader applicability is the fact that the conclusions are also stochastic, and therefore *less specific* or less *informative* than a deterministic one.

In this thesis, we describe the environment of the queues and therefore the queues themselves in a stochastic mathematical framework. This means a certain probability is assigned to every possible event. For instance, instead of specifying whether a user will generate a packet or not, we simply bypass the complex analysis of the causes that would lead to either event by saying there is a probability p that a packet is generated and a probability $1-p$ that there is not[4]. The formalised mathematical laws that allow to draw further conclusions from such stochastic assumptions is called *probability theory*, of which we briefly review some elements here. In doing so, we adhere to the axiomatic setting as first proposed by A.N. Kolmogorov in 1933 [117] and strictly limit our discussion to variables and processes that are discrete in both time and space. We particularly focus on the versatility of using *probability generating functions* (pgfs) to represent distributions and on the concept of *Markov chains*, both of which are key elements to the general methodology in this thesis. The main reference on probability theory and stochastic processes is still [82]. An overview of stochastic processes with specific importance to telecommunication can be found in [142].

1.2.1 Discrete random variables

We assume the reader has at least an intuitive notion of what is meant by a random experiment, an event, the probability $\mathrm{Prob}[A]$ of an event A, and of concepts like the conditional probability $\mathrm{Prob}[A|B]$, joint probability $\mathrm{Prob}[A,B]$ and basic laws of probabilistic reasoning.

A *random variable* a is a measurable[5] function that maps all possible outcomes of an experiment to a set of 'observable' values V. The set V is often taken to be the set $\mathbb{R}$ of real numbers, in which case a is called a *continuous* random variable. For example, an experiment could be to take a pair of dice and roll them over a table. The outcome or 'sample' of this experiment is for the dice and the table to have assumed a certain state ω. The set of all possible outcomes is denoted as Ω and called the *universe*. Consider then the function d which maps all possible states ω of the universe Ω to a number $d(\omega) \in \mathbb{R}$ such that in case of an outcome ω, the distance between the two dice measured in centimeters is $d(\omega)$.

If on the other hand, V is the set of integer numbers only, then a is called a *discrete* random variable. For example, doing the same experiment as above, now define another function a_T such that $a_T(\omega)$ is always equal to the number of dots on the top surfaces of the dice in case the outcome is ω. Clearly, it is possible to construct many more random variables that relate the outcomes of this particular experiment to real or integer numbers. In fact, doing so sensibly, is a way of gathering statistical information about the experiment. However, instead of rolling dice, we rather study

[4] Whether this is an adequate assumption is another issue, see Sec. 1.3.1.

[5] Probability theory, like most mathematical disciplines, is rooted in measure theory and set theory. The requirement of measurability is to avoid certain inconsistencies in the theory, see e.g. [100].

queues whereby, as we will discuss further, all quantities of interest are discrete. They are also nonnegative, i.e. we assume that all discrete random variables take values in the set $\mathbb{N}$ of natural numbers, unless explicitly stated otherwise. Therefore, from this point we may refer to a discrete nonnegative random variable simply as a 'variable' or 'quantity'.

Distribution of a discrete random variable

A discrete random variable a is characterised by its *distribution*, which basically is any representation that quantifies the probability Prob$[a \in W]$ for all subsets W of $V = \mathbb{N}$. One way to represent the distribution of a is by the sequence

$$p_a(n) \triangleq \text{Prob}[a = n]\,, \quad n = 0, 1, 2, \dots\,,$$

which gives us the probability that the outcome of the experiment will be mapped to the number n, or in other words, the probability of observing the event '$a = n$'. This sequence of probabilities $p_a(n)$ is called the *probability mass function* (pmf) of a, for which

$$p_a(n) \geqslant 0, \quad \forall n \geqslant 0\,, \tag{1}$$

$$\sum_{n=0}^{+\infty} p_a(n) = 1\,, \tag{2}$$

must hold[6]. If a has some finite probability of assuming the value n (i.e. $p_a(n) > 0$), then we call n a *mass point* of a. The distribution of a can also be given as its *cumulative distribution function* (CDF) $F_a(n)$ defined as

$$F_a(n) \triangleq \text{Prob}[a \leqslant n] \;=\; \sum_{i=0}^{n} \text{Prob}[a = i] \;=\; \sum_{i=0}^{n} p_a(i)\,,$$

for which obviously, $\lim_{n\to\infty} F_a(n) = 1$ due to (2). The CDF gives the probability of a not *exceeding* a certain value n, while the pmf returns the probability of a being equal to n. Both functions represent the distribution uniquely and the CDF can always be derived from the pmf and vice versa. In the next section, we will introduce yet another distributional form for discrete random variables.

An interesting thing about random variables is the fact that *new* random variables can be defined out of old ones with ease. Specifically, for any measurable function f from V to $f(V)$, it is clear that $a' = f(a)$ is again a random variable with possible values in $f(V)$. For our purposes, it is sufficient to consider functions f from $\mathbb{N}$ to $\mathbb{N}$ only, for which the measurability is implied. The function f also provides the distribution of a'. For its pmf for instance, we have

$$p_{a'}(n) \;=\; \text{Prob}[f(a) = n] \;=\; \sum_{i=0}^{+\infty} \text{Prob}[a = i]\, \delta(f(i) = n)\,,$$

[6]Random variables for which (2) does not hold are called *deficient* (see [82]) and are used to represent quantities that are equal to ∞ with finite probability.

where the function δ is 1 if its argument is true and 0 otherwise. In general, a discrete (nonnegative) distribution is determined by a countable sequence of real numbers (the probabilities) for which the only constraints are (1)–(2). Nevertheless, it is sometimes useful to characterise a distribution with one or a few numbers only. An important operator in this regard is the *expected value* $\mathrm{E}[\cdot]$ of a random variable, which for $a' = f(a)$ is defined as

$$\mathrm{E}[a'] = \sum_{n=0}^{+\infty} n\, p_{a'}(n) \tag{3}$$

$$= \mathrm{E}[f(a)] = \sum_{n=0}^{+\infty} f(n)\, p_a(n)\,. \tag{4}$$

This operator is linear, i.e. $\mathrm{E}[ca] = c\mathrm{E}[a]$ for any $c \in \mathbb{R}$, but generally does *not* commute with f: $\mathrm{E}[f(a)] \neq f(\mathrm{E}[a])$. Note that if we have a functional relation $a' = f(a)$, then $\mathrm{E}[a']$ can be calculated either by using the pmf of a' or that of a. This is a fundamental property of the expected value operator that can be useful in case $p_{a'}(n)$ is hard to obtain explicitly and $p_a(n)$ is not. The operator $\mathrm{E}[\cdot]$ can be used to obtain the *moment* and the *central moment* of order k as

$$m_a(k) \triangleq \mathrm{E}[a^k]\,, \qquad \text{and} \qquad c_a(k) \triangleq \mathrm{E}\big[\big(a - \mathrm{E}[a]\big)^k\big]\,, \tag{5}$$

respectively. The moments reveal specific information about the *shape* of the distribution. For instance, $m_a(1) = \mathrm{E}[a]$ is the value around which the pmf of a is 'centered' and is also the mean value of the observed values for a if the experiment is repeated many times. The central moments on the other hand, give a measure for how much the observed values deviate around the center. Of particular importance is the second order central moment

$$\mathrm{Var}[a] \triangleq c_a(2) = \mathrm{E}\big[\big(a - \mathrm{E}[a]\big)^2\big]\,, \qquad \text{and} \qquad \mathrm{Dev}[a] \triangleq \sqrt{\mathrm{Var}[a]}\,,$$

which are respectively called the *variance* and the *standard deviation* of a. For some distributions, not all moments are finite. However, *if* all moments exist, the sequence $m_a(k)$ (or $c_a(k)$), $k \geqslant 0$, uniquely determines the distribution of a as well.

Joint distribution of multiple variables

As in our dice rolling example above, it is perfectly possible to consider *multiple* discrete random variables $a_1, a_2, \ldots, a_N$ from the same experiment. Let us only consider a_1 and a_2 here, since the extension to $N > 2$ is straightforward. Considering two random variables is sometimes seen as conducting a 'combined' experiment, although there is in fact only *one* experiment, but that is observed in two different ways. As such, it would be better to talk about a *combined observation*, i.e. if the outcome of the experiment is ω, the observations are the numbers $a_1(\omega)$ and $a_2(\omega)$, which can be seen as a single random variable $\boldsymbol{a} = \{a_1, a_2\}$ taking values in $V = \mathbb{N}\times\mathbb{N}$. In general, a combined variable $\boldsymbol{a} \triangleq \{a_1, a_2, \ldots, a_N\}$ is said to be *multivariate* and the set $V = \mathbb{N}^N$ of observable values is qualified as *multidimensional*. As before, the distribution of the discrete variable $\boldsymbol{a}$ can be given by stating the probability of all events '$\boldsymbol{a}=v$' for all $v \in V$, which here means:

$$p_{\boldsymbol{a}}(v) \;=\; p_{a_1 a_2}(i,j) \triangleq \text{Prob}[a_1 = i, a_2 = j]\,, \quad i,j \geqslant 0\,. \tag{6}$$

The function $p_{a_1 a_2}(i,j)$ is called the *joint* probability mass function of a_1 and a_2. The pmf of either a_1 or a_2 alone can be obtained from their joint pmf as e.g.

$$p_{a_1}(i) \;=\; \sum_{j=0}^{+\infty} p_{a_1 a_2}(i,j)\,.$$

However, the converse is not true. It is not generally possible to obtain the joint distribution (6) from the 'marginal' pmfs $p_{a_1}(i)$, $i \geqslant 0$ and $p_{a_2}(j)$, $j \geqslant 0$.

Sometimes, when conducting an experiment, the outcome is only partially revealed. Instead of knowing the specific outcome ω^* and hence the value of all random variables defined on this experiment, we may only be told that an event B has occurred, i.e. that $\omega^* \in B$. Given this knowledge, this may affect the *a posteriori* distribution of the random variables involved. Formally, we define the *conditional* mass function of a given a certain event B as

$$p_{a|B}(n) \triangleq \text{Prob}[a = n | B] = \frac{\text{Prob}[a = n, B]}{\text{Prob}[B]}\,, \quad n \geqslant 0. \tag{7}$$

Very often, the event B on which we condition is that of another variable having a certain value, for example $B =$ '$a_2 = j$'. If so, we denote the conditional pmf of a_1 as

$$p_{a_1|B}(i) \;=\; p_{a_1|a_2}(i|j) \;=\; \text{Prob}[a_1 = i | a_2 = j]\,. \tag{8}$$

For example, in the dice rolling experiment, let a_1 and a_2 be the number of dots on the top surface of each die respectively and let $a_T = a_1 + a_2$ be the total number thrown by the player. Suppose then that after throwing, only the second die can be observed, which reads e.g. $a_2 = 1$. Clearly, this condition alters the distribution of a_T. It would for instance be impossible now to have a total a_T larger than 7, whereas in the *un*conditional case this is $\text{Prob}[a_T > 7] = 15/36$. In other words, taking into account that the event '$a_2 = j$' occurred, the distribution of a_T is given by $p_{a_T|a_2}$ rather than p_{a_T}.

As with the pmf, conditioning on an event may also affect the expected value of a random variable. We define the *conditional* expected value of a given an event B as

$$\text{E}[a|B] \;=\; \sum_{n=0}^{+\infty} n\text{Prob}[a = n | B]\,.$$

Considering two variables a_1 and a_2 again, the law of *total expectation* then simply follows from the law of total probability:

$$\text{E}[a_1] \;=\; \sum_{i=0}^{+\infty} i\,\text{Prob}[a_1 = i] \;=\; \sum_{i=0}^{+\infty} i \sum_{j=0}^{+\infty} \text{Prob}[a_1 = i, a_2 = j]$$

$$= \sum_{j=0}^{+\infty} \Big(\sum_{i=0}^{+\infty} i \,\mathrm{Prob}[a_1 = i | a_2 = j] \Big) \mathrm{Prob}[a_2 = j]$$

$$= \sum_{j=0}^{+\infty} \mathrm{E}[a_1 | a_2 = j] \,\mathrm{Prob}[a_2 = j] \;=\; \mathrm{E}\left[\, \mathrm{E}[a_1 | a_2 = j]\, \right]_{a_2} , \tag{9}$$

which says that $\mathrm{E}[a_1]$ can be obtained by first conditioning a_1 on '$a_2 = j$' and then averaging out over these events.

Statistical independence

Nevertheless, knowledge of one variable's value need not *always* affect the distribution of another variable. Two variables a_1 and a_2 are said to be *statistically independent* if and only if

$$p_{a_1 a_2}(i, j) \;=\; p_{a_1}(i)\, p_{a_2}(j) \quad \text{for all } i, j \geqslant 0 \,. \tag{10}$$

So statistical independence of two random variables is entirely determined by their joint distribution. There is an intuitive interpretation to the concept of independency. Note that if a_1 and a_2 are independent, (10) together with (7) implies that

$$p_{a_1|a_2}(i|j) \;=\; p_{a_1}(i) \,, \qquad \text{for all } i, j \geqslant 0 \,.$$

So the fact that a_2 is known, is *irrelevant* to the distribution of a_1 and vice versa. The intuitive interpretation of this invariance is that the two variables do not contain any 'information' about each other. For example, the numbers a_1 and a_2 thrown with each of the dice will be statistically independent (at least if the player and the dice are honest), but e.g. a_1 and a_T will clearly not.

As in the case of a single random variable, a function $f(\boldsymbol{a})$ of two random variables $\boldsymbol{a} = \{a_1, a_2\}$ is again a random variable, of which the pmf is given by

$$p_{f(\boldsymbol{a})}(n) \;=\; \sum_{v \in V} p_{\boldsymbol{a}}(v)\, \delta(f(v) = n) \;=\; \sum_{i=0}^{+\infty} \sum_{j=0}^{+\infty} p_{a_1 a_2}(i, j)\, \delta(f(i, j) = n) \,, \tag{11}$$

for $n \geqslant 0$, assuming that the range of f is also $\mathbb{N}$. It is often difficult to obtain (11) explicitly. Consider for example the sum $a_T = a_1 + a_2$, then

$$p_{a_T}(n) \;=\; \sum_{i=0}^{+\infty} \sum_{j=0}^{+\infty} p_{a_1 a_2}(i, j)\, \delta(i + j = n)$$

$$= \sum_{i=0}^{n} p_{a_1 a_2}(i, n - i) \;=\; \sum_{i=0}^{n} p_{a_1|a_2}(i | n - i) p_{a_2}(n - i) \,.$$

If a_1 and a_2 are statistically independent, this can be further reduced to

$$p_{a_T}(n) \;=\; \sum_{i=0}^{n} p_{a_1}(i)\, p_{a_2}(n - i) \triangleq p_{a_1} * p_{a_2}(n) \,, \tag{12}$$

which is called the *convolution* of p_{a_1} and p_{a_2}. So even for a simple function as the sum of two independent variables, a lot of computation is required to obtain the resulting pmf.

Correlation

The expected value operator can also be used to obtain scalar characteristics of the *joint* distribution of a_1 and a_2, i.e.

$$\mathrm{E}[f(\boldsymbol{a})] \;=\; \sum_{v \in V} f(v)\, p_{\boldsymbol{a}}(v) \;=\; \sum_{i=0}^{+\infty}\sum_{j=0}^{+\infty} f(i,j) p_{a_1 a_2}(i,j)\,. \tag{13}$$

This can be used to produce several interesting scalar characteristics of the joint distribution, much like the moments in (5). The most important measure in this regard is the *covariance* of two random variables defined as

$$\mathrm{Cov}[a_1, a_2] \triangleq \mathrm{E}\left[\, \big(a_1 - \mathrm{E}[a_1]\big)\big(a_2 - \mathrm{E}[a_2]\big) \,\right] \;=\; \mathrm{E}[a_1 a_2] - \mathrm{E}[a_1]\mathrm{E}[a_2]\,,$$

and quantifies how much the joint distribution fluctuates around the mean in both dimensions. Scaling the covariance as

$$\phi_{a_1 a_2} \triangleq \frac{\mathrm{Cov}[a_1, a_2]}{\mathrm{Dev}[a_1]\mathrm{Dev}[a_2]}\,,$$

gives the *coefficient of correlation* between a_1 and a_2, which is a real number between -1 and 1. The correlation coefficient indicates how well both distributions follow the same linear trend. If $\phi\!=\!0$, no such general trend can be found, whereas if $|\phi|\!=\!1$ then all mass points of $\{a_1, a_2\}$ in the plane with coordinates a_1 and a_2 are on the same line. The sign of ϕ indicates whether the linear trend is increasing or decreasing. Two variables are said to be *correlated* if their covariance or correlation coefficient differs from 0. Correlation between two variables indicates a certain ‘overall’ dependence between their distributions, although correlation and statistical dependence are *not* the same. If two variables are correlated, they must be statistically dependent, but dependency does not imply correlation. Also, two variables a_1 and a_2 that are independent, are always uncorrelated, since

$$\mathrm{E}[a_1\, a_2] \;=\; \sum_{i=0}^{+\infty}\sum_{j=0}^{+\infty} i\, j\, p_{a_1 a_2}(i,j) \;=\; \sum_{i=0}^{+\infty} i\, p_{a_1}(i) \sum_{j=0}^{+\infty} j\, p_{a_2}(j) \;=\; \mathrm{E}[a_1]\,\mathrm{E}[a_2]\,,$$

but again, the converse is not true in general.

1.2.2 Probability generating functions

One of the goals of this thesis is to advocate the use of *probability generating functions* as the preferred way of representing a discrete distribution. As we will demonstrate, in the analysis of stochastic systems there are some significant advantages to manipulating distributions in the form of their pgfs rather than their pmfs.

Let a be a discrete, nonnegative and nondeficient random variable with pmf $p_a(n)$, $n \geqslant 0$. For ease of notation, let us henceforth denote the probability of a having value n as

$$a(n) \triangleq p_a(n) \;=\; \mathrm{Prob}[a\!=\!n]\,, \qquad n \geqslant 0\,. \tag{14}$$

The probability generating function $A(z)$ of a is defined as the function

$$A(z) \;=\; \mathrm{E}[z^a] \;=\; \sum_{n=0}^{+\infty} a(n)\, z^n\,. \tag{15}$$

This is the so-called z-transform of the sequence (14) to a function $A(z)$ in the complex plane $z \in \mathbb{C}$, valid for all z for which the power series (15) converges. Observe that the transform is one to one, i.e. with each pmf corresponds exactly one pgf and vice versa. The distribution of a is therefore defined uniquely by its pgf $A(z)$ and it is possible to conduct all stochastic inference in terms of probability generating functions instead of mass functions. Manipulating distributions in the form of pgfs, or in the *transformation domain* as this is called, requires an entirely different mathematical approach than working in the probability domain, i.e. using the pmf or CDF. For example, pgfs are functions in the complex plane, so a basic knowledge of complex function theory is needed, see e.g. [36, 86, 99]. However, as the coefficients in (15) are real and nonnegative, it is seen from the definition that for real $z \geqslant 0$, the pgf $A(z)$ is real and nonnegative as well. By somewhat loosening the mathematic rigor, most of the properties that are important in practice can be deduced equally well by regarding $A(z)$ as a real function.

Immediate properties

Some immediate properties of the pgf $A(z)$ are the following:

- *Normalisation*
  The coefficients $a(n)$ in definition (15) are the probabilities that constitute the mass function $p_a(n)$, for which the normalisation condition (2) holds. Taking $z = 1$ thus yields

  $$A(1) \;=\; \sum_{n=0}^{+\infty} a(n) \;=\; 1\,,$$

  which is the normalisation condition that holds for any (proper) pgf. We will often use this property to determine a final unknown parameter of a distribution.
- *Bounded and analytic*
  The function (15) is a power series in z and therefore *analytic* within a certain region of convergence $|z| < R$. This region contains at least the open unit disc $\mathring{D} = \{z : |z| < 1\}$, i.e. $R \geqslant 1$ since for all $|z| \leqslant 1$ we have

  $$|A(z)| \leqslant \sum_{n=0}^{+\infty} |a(n)|\, |z|^n \leqslant \sum_{n=0}^{+\infty} a(n) \;=\; 1\,,$$

  which also means that $A(z)$ is *bounded* in the closed unit disc $D = \{z : |z| \leqslant 1\}$. The fact that $A(z)$ is analytic in $\mathring{D}$ implies that it is *differentiable* up to any order in that region. It also implies, by definition, that $A(z)$ can have no singularities in $\mathring{D}$, an observation that can be used to determine unknown parameters, just like the normalisation condition. Moreover, most pgfs that occur in practice are analytic in a region that extends at least a bit beyond D and which includes the point $z = 1$.

We assume this is the case for all pgfs in this thesis, such that all the derivatives $A'(z), A''(z), \ldots, A^{(n)}(z), \ldots$ with

$$A^{(n)}(z) \;=\; \frac{\mathrm{d}^n}{\mathrm{d}z^n} A(z)\,, \qquad n > 0\,,$$

are defined uniquely on the unit disc D.

- *The probability generating property*
This is the property to which pgfs owe their name. The probabilities $a(n)$ can always be 'generated' from the function $A(z)$ by consecutive differentiation to its argument z and evaluating for $z\!=\!0$:

$$a(n) \;=\; \frac{1}{n!}\left[A^{(n)}(z)\right]_{z=0} . \tag{16}$$

In principle, this provides a procedure for inverting $A(z)$ back to the probability domain, although it usually is an impractical one. A useful consequence of (16) is however that

$$A(0) \;=\; a(0) \;=\; \mathrm{Prob}[a\!=\!0]\,.$$

- *Moment generating property*
The derivatives of $A(z)$ can also be evaluated in the point $z = 1$. It is seen from (15) that this yields

$$\left[A^{(n)}(z)\right]_{z=1} \;=\; \mathrm{E}[a\,(a\!-\!1)\cdot\,\ldots\,\cdot(a\!-\!n\!+\!1)]\,, \tag{17}$$

which are called the *factorial* moments of a. It is possible to obtain the moments $m_a(k)$ and central moments $c_a(k)$ in (5) from these factorial moments in a recursive way. For the first and second order moments, we have for instance:

$$\begin{aligned} \mathrm{E}[a] &= A'(1)\,, \\ \mathrm{E}[a^2] &= A''(1) + A'(1)\,, \\ \mathrm{Var}[\,a\,] &= \mathrm{E}[a^2] - \mathrm{E}[a]^2 = A''(1) - \big(A'(1)\big)^2 + A'(1)\,. \end{aligned}$$

Joint, conditional and partial probability generating functions

It is also possible to express the joint distribution of multiple random variables a_1, $a_2, \ldots, a_N$ in the z-transform domain. Again considering only the case $N = 2$, the *joint probability generating function* of a_1 and a_2 is a complex function in two complex arguments defined as the two-fold power series

$$A(z_1, z_2) \;=\; \mathrm{E}[z_1^{a_1}\cdot z_2^{a_2}] \;=\; \sum_{i=0}^{+\infty}\sum_{j=0}^{+\infty} a(i,j)\cdot z_1^i \cdot z_2^j\,, \tag{18}$$

where we have introduced the notation

$$a(i,j) \;=\; p_{a_1 a_2}(i,j) \;=\; \mathrm{Prob}[a_1\!=\!i, a_2\!=\!j]\,,$$

for the joint pmf of both variables. We will often consider joint pgfs with more than two and even an infinite number of arguments. In principle, joint pgfs have the same properties as one-dimensional pgfs. They are bounded in the region of their domain where both $|z_1| \leqslant 1$ and $|z_2| \leqslant 1$. They normalise to 1 for $z_1 = z_2 = 1$ and allow for the marginal pgfs $A_1(z)$ and $A_2(z)$ of respectively a_1 and a_2 to be obtained easily,

$$A_1(z_1) \;=\; A(z_1, 1)\,, \qquad \text{and} \qquad A_2(z_2) \;=\; A(1, z_2)\,. \tag{19}$$

Also, by taking suitable partial derivatives (PDs) of $A(z_1, z_2)$ evaluated in $z_1 = z_2 = 1$, the factorial moments and several kinds of cross-moments can be obtained again,

$$\begin{aligned} \mathrm{E}[a_1] &= A_1'(1) = \frac{\partial}{\partial z_1} A(1,1) & \mathrm{E}[a_2] &= A_2'(1) = \frac{\partial}{\partial z_2} A(1,1)\,, \\ \mathrm{E}[a_1(a_1 - 1)] &= A_1''(1) = \frac{\partial^2}{\partial z_1^2} A(1,1) & \mathrm{E}[a_2(a_2 - 1)] &= A_2''(1) = \frac{\partial^2}{\partial z_2^2} A(1,1)\,, \end{aligned}$$

and

$$\mathrm{E}[a_1\, a_2] \;=\; \frac{\partial^2}{\partial z_1 \partial z_2} A(1,1)\,.$$

If two variables are statistically independent, this can be readily observed from their joint pgf. Indeed, the equivalent of (10) in the z-domain is

$$\begin{array}{c} a_1 \text{ and } a_2 \text{ are} \\ \text{statistically independent} \end{array} \quad \Longleftrightarrow \quad A(z_1, z_2) \;=\; A_1(z_1) \cdot A_2(z_2)\,,$$

for all z_1 and z_2 in some nonempty open region of $\mathbb{C}$ where (18) converges.

The *conditional* pgf of a given that a certain event B happened is simply the transform of the conditional pmf (7),

$$A_{\text{cond } B}(z) \triangleq \sum_{n=0}^{+\infty} z^n\, p_{a|B}(n)\,.$$

Sometimes it is useful to 'rescale' $A_{\text{cond } B}(z)$ to the probability of B, i.e.

$$\begin{aligned} A_{\text{part } B}(z) &\triangleq A_{\text{cond } B}(z)\, \mathrm{Prob}[B] \\ &= \sum_{n=0}^{+\infty} z^n\, p_{a|B}(n)\, \mathrm{Prob}[B] \;=\; \sum_{n=0}^{+\infty} z^n\, \mathrm{Prob}[a = n, B]\,. \end{aligned}$$

This is called a *partial* pgf because it generates the probabilities of a joint with B. Note that

$$A_{\text{part } B}(z) + A_{\text{part } \bar{B}}(z) \;=\; A(z)\,,$$

where $\bar{B}$ is the complementary event of B. Obviously, $A_{\text{part } B}(z)$ does not normalise to 1 but to $\mathrm{Prob}[B]$, indicating the 'size' of its share in the total $A(z)$. Note also that a pgf can be partial and conditional at the same time, i.e. the transform of some probability sequence $\mathrm{Prob}[a = n, C|B]$. Equivalent definitions for multidimensional partial and conditional transforms are possible as well.

Stochastic inference in the transformation domain

As said already, when the distribution of a random variable has to be determined that is some function of *other* random variables, it is often advantageous to use generating functions rather than mass functions. We illustrate this assertion with two examples:

► *Sum of independent variables*
Suppose that a_1 and a_2 are independent. To obtain the pmf of the sum $a_T = a_1 + a_2$, we have to use the discrete convolution as in (12). In the z-domain however, we simply have that

$$A_T(z) \;=\; \mathrm{E}[z^{a_T}] \;=\; \mathrm{E}[z^{a_1+a_2}] \;=\; \mathrm{E}[z^{a_1}]\,\mathrm{E}[z^{a_2}] \;=\; A_1(z)\,A_2(z)\,.$$

So the pgf of a sum of independent variables is obtained simply by multiplying the pgfs of each term in the sum, which is a lot easier than calculating convolutions in the probability domain.

► *Stochastic sums*
Let us now consider the sum $a_T = a_1 + a_2 + \ldots + a_N$ in which all terms are independent and identically distributed (iid) with common pgf $A(z)$. If N, the number of terms in this sum is deterministic then we have just seen that $A_T(z) = (A(z))^N$. Suppose however that N is itself a random variable, independent of the terms in the sum and with pgf $N(z)$. The pgf $A_T(z)$ can then be obtained by using (9), the law of total expectation:

$$\begin{aligned} A_T(z) \;&=\; \mathrm{E}[z^{a_T}] \;=\; \mathrm{E}\left[\,\mathrm{E}[z^{a_1+\ldots+a_N}|N]\,\right]_N \;=\; \mathrm{E}\left[\mathrm{E}[z^{a_1}|N]\ldots\mathrm{E}[z^{a_N}|N]\right]_N \\ &=\; \mathrm{E}\left[\mathrm{E}[z^{a_1}]\ldots\mathrm{E}[z^{a_N}]\right]_N \;=\; \mathrm{E}\left[\,\big(A(z)\big)^N\,\right]_N \;=\; N\big(A(z)\big)\,. \end{aligned}$$

So the pgf of a stochastic sum of iid terms is also very easy to obtain as the pgf of the number of terms with the common pgf of the terms as its argument.

In view of the fact that in stochastic models, a *lot* of random variables are defined as the sum of independent quantities, these calculation rules prove to be very useful.

Inversion: back to the probability domain

Unfortunately, stochastic inference in the transformation realm has disadvantages as well. The most important drawback is likely the difficulty of translating the results back to the probability domain. After obtaining the pgf $A(z)$ of a certain stochastic variable a of interest, we can calculate its moments with (17). But often we are interested in the pmf $a(n) = \mathrm{Prob}[a = n]$ as well. If we introduce the notation $[z^n]F(z)$ to indicate the coefficient of z^n in the power series expansion of the function $F(z)$ around the origin, we thus have that $a(n) = [z^n]A(z)$.

As a possible solution, we already mentioned the probability generating property (16). If the available expression for $A(z)$ is not too elaborate, this can be used to calculate the probabilities $a(0), a(1), \ldots$ up to $a(n)$ for some small n. The complexity of the symbolic derivation however, increases exponentially with growing n such that this method is impractical in most cases. Another theoretical approach is to use the *inverse* z-transformation formula, see [36, 99]. From (15) it is seen that $a(n)$ is the coefficient of z^{-1} in the series expansion of $A(z)z^{-n-1}$ around the pole $z=0$. The Cauchy residue theorem then yields

$$a(n) \; = \; [z^n]A(z) \; = \; \frac{1}{2\pi\jmath} \oint_{C_r} A(z)z^{-n-1}\mathrm{d}z \; = \; \mathrm{Res}_{z=0}\left[A(z)z^{-n-1}\right] \; , \tag{20}$$

where C_r is a contour around the origin lying within the region of convergence of $A(z)$ and $\jmath$ the complex imaginary unit $\sqrt{-1}$. In practice, this does not help us further since we need the subsequent differentiations again to calculate the residue in (20).

Nevertheless, the contour integration formula *does* form the starting point for some methods that can very accurately *approximate* $a(n)$ for large n, the so-called *tail distribution* of the variable a. These asymptotic methods are based on identifying the position and the nature of the *singularities* of $A(z)$, i.e. the points in $\mathbb{C}$ where $A(z)$ is not analytic. The single contour C_r in (20) can be extended to encompass the whole complex plane, except for the singularities of $A(z)$. The contribution along the contour at infinity drops to 0 if n is large enough, leaving only contributions along some contours around the singularities. How these contributions must be calculated, then very much depends on whether the singularities in question are isolated or not, whether they are essential singularities, poles, or part of a branch cut. For the sake of argument, suppose that $A(z)$ has M isolated singularities in the points $z=z_i$, $i=1,\ldots,M$, then

$$a(n) \; = \; -\frac{1}{2\pi\jmath} \sum_{i=0}^{M} \oint_{C_i} A(z)z^{-n-1} \, , \tag{21}$$

with C_i a contour circling around z_i only. If all of these integrals could be evaluated, then this would yield $a(n)$ exactly. For large n however, it can be shown that the terms of (21) with contours around the singularities with *smallest modulus* will dominate the other terms. These singularities are therefore called the *dominant* singularities, which are positioned on a circle around the origin with radius R, the radius of convergence of $A(z)$. This is due to Pringsheim's theorem, see [85, 186].

Theorem 1 (Pringsheim's theorem)**.** *A complex function $F(z)$ whose power series expansion at $z=0$ has a finite radius of convergence R, necessarily has a singularity on the boundary of its disc of convergence $|z|=R$. If the coefficients $f(n)=[z^n]F(z)$ in this expansion are nonnegative, the point $z=R$ is a singularity of $F(z)$.*

Deriving the asymptotic behaviour of $a(n)$ by studying the local behaviour of the function $A(z)$ around its dominant singularities has been named *singularity analysis*, see e.g. [74, 84, 85, 157] and their references. But what if n is too large to use the probability generating property (16), and too small for the asymptotic approximation to be accurate enough? If an analytic expression for $a(n)$ is desired, one could try to obtain additional terms in (21), preferably the contributions of the contours around singularities close to the dominant ones. This can sometimes increase the accuracy of the approximation for lower n. However, if the analytic approach becomes unfeasible, one can always resort to numerical methods, see e.g. [20, 111] and the appendix of [123] for a discussion.

In each of the following chapters, we will study the asymptotic decay or 'tail behaviour' of the random variables involved by inspecting the dominant singularity. For all of the quantities concerned, it turns out that there is a single dominant pole that determines the tail distribution.

A problem related to the inversion of pgfs is the following. Given an arbitrary complex function $A(z)$ such that $A(1)=1$, is there a simple criterium to decide whether $A(z)$ is a proper pgf or not? In other words, do the coefficients in the series expansion of $A(z)$ represent a probability distribution? Note that the question is not trivial, the fact that the function is normalised at $z=1$ does *not* necessarily make it a pgf (take e.g. $A(z)=z^2+z-1$). To the best of our knowledge, there seems to be no easy way to check directly in the transformation domain whether all coefficients in the expansion are nonnegative. So in general, without inverting $A(z)$, we cannot be sure that it is a pgf. Luckily for us however, we usually *know* that the functions we obtain must be proper pgfs, because they result from stochastic inference rules.

In general, one can define the power series (15) for *any* sequence of finite nonnegative numbers $f(n)$, which yields a function $F(z)$ that is analytic at least in some region around $z=0$. These functions are simply called generating functions instead of 'probability' generating functions, since $F(1)$ does not necessarily converge. And even if it does converge, it is not always to a number $0 \leqslant F(1) \leqslant 1$ as would be the case for pgfs or partial pgfs. Such functions are useful in many computer science disciplines such as combinatorics and graph theory, see [85] again and [116].

1.2.3 Discrete-time stochastic processes

Essentially, a *random* or *stochastic* process is just a special kind of random variable. We demonstrate the natural relation between the two concepts as follows. Consider the random variable a^* and recall that this is a function mapping the outcome ω of an experiment to a set V^* of observed values:

$$a^* : \Omega \mapsto V^* : \omega \rightarrow v^* \,.$$

The unusual thing with this variable a^* is that its 'values', i.e. the elements of V^*, are themselves functions over some *index set* T, which is usually $T=\mathbb{R}$ interpreted as the points in linear time. So with every possible outcome ω of the experiment now corresponds a single time-dependent function $v^*=v(t)$, i.e.

$$a^* = a_t : \Omega \mapsto V(T) : \omega \rightarrow v(t)\,,$$

where we have denoted the set of all functions from T to V as $V(T)$. The fact that a_t is called a *process* now instead of a variable, is only to account for the time-dependency of the 'values' $v(t)$. In this new framework, a particular $v(t) \in V(T)$ is called a *realisation* of the process a_t and the set $V(T)$ of all possible realisations is the *ensemble* of a_t. The set V, i.e. the range of the realisations $v(t)$, is called the *state space* of a_t.

It is customary to classify stochastic processes according to the nature of the index set T on the one hand, and the state space V on the other. If the index set T is a non-countable subset of $\mathbb{R}$, then a_t is called a *continuous-time* stochastic process. For example, in the dice-rolling experiment of the previous section, we could extend the variable d which is the distance between the dice as they lie still on the table, to a process $d^*=d(t)$ that records the distance as the dice *are rolling* over the table (note that $\lim_{t\to\infty} d_t = d$, then). If on the other hand T is countable, a_t is said to be a *discrete-time* process. We indicate this by using the index $k \in \mathbb{N}$ instead of t. In this case, the process a_k can also be regarded as a *sequence* of separate random variables

$$a_0, a_1, a_2, \dots\,, \tag{22}$$

defined over the same probability space. Note that there is little difference between a multivariate random variable $\boldsymbol{a} = \{a_1, a_2, \dots, a_N\}$ as in Sec. 1.2.1 and a discrete-time stochastic process a_k with $T = \{k : 1 \leqslant k \leqslant N\}$. Both represent a collection of N variables with values in V and whether they are to be seen as a multivariate or a stochastic process is just a matter of interpretation. Besides the domain T of the realisations $v(t)$, their range V classifies the process as well. If V is non-countable, then a_t (or a_k) is called a *continuous* process, whereas it is *discrete* if V is countable. In the latter case, the process is often called a *chain*.

In what follows, we always assume that $T = \mathbb{N}$ *and* $V = \mathbb{N}$, so we only consider stochastic processes that are discrete in both space and time. Henceforth, if we use the term stochastic 'process', we mean a discrete sequence of discrete-valued random variables in the sense of (22).

Statistical characterisation of a discrete-time chain

We can also extend the concept of 'probability distribution' to stochastic processes. The nth order statistical properties of a process a_k are characterised by the joint probability mass function

$$p_{a_k}(i_1, i_2, \dots, i_n; k_1, k_2, \dots, k_n) \;=\; \text{Prob}[a_{k_1} = i_1, a_{k_2} = i_2, \dots, a_{k_n} = i_n]\,, \tag{23}$$

for all possible choices of the n indices $k_1, \dots, k_n$. In principle, to fully characterise a general stochastic process, we need the nth order mass functions for all $n \geqslant 1$ and for all possible choices of $k_1, k_2, \dots, k_n$. However, it often suffices to have the first and second-order statistics only.

A process a_k is *stationary* in the strict sense if it is statistically identical to $a_{k+\ell}$ for all $\ell = \dots, -1, 0, 1, \dots$, i.e. if for all $i_1, \dots, i_n$ and $k_1, \dots, k_n$ we have that

$$\begin{aligned} &p_{a_k}(i_1, i_2, \dots, i_n; k_1 + \ell, k_2 + \ell, \dots, k_n + \ell) \\ &\qquad = p_{a_k}(i_1, i_2, \dots, i_n; k_1, k_2, \dots, k_n)\,, \quad \text{for all } n. \end{aligned} \tag{24}$$

Stationarity implies that the nth order mass functions only depend on the relative distances between the indexes k_j and not on their absolute positions on the time axis. A process is called stationary in the wide sense if (24) holds for the first and second-order distributions ($n = 1, 2$). Many of the processes in the further chapters are not stationary, but we assume them to be *asymptotically* stationary, in either the strict or wide sense. This means that

$$\lim_{\ell\to\infty} p_{a_k}(i_1, i_2, \dots, i_n; k_1 + \ell, k_2 + \ell, \dots, k_n + \ell)\,, \tag{25}$$

exists and becomes independent of ℓ. Following such processes over time, once the distribution (25) no longer changes with ℓ, the process is said to have reached *steady* *state* or stochastic *equilibrium*[7].

For (asymptotically) stationary processes we can define the so called *stationary distribution* given by the mass function

$$p_a(i) \;=\; \lim_{k\to\infty} \mathrm{Prob}[a_k\!=\!i]\,,$$

which is independent of k. This is the pmf of a 'generic' random variable a which has the stationary distribution of the process a_k during equilibrium. Later we shall often drop the index k in the notation to indicate the random variable that is the 'steady-state version' of the process in question. Another useful characterisation of stationary processes is their *autocorrelation function* (ACF) defined as

$$R(\ell) \;=\; \lim_{k\to\infty} \mathrm{E}\left[\,\big(a_k - \mathrm{E}[a_k]\big)\big(a_{k+\ell} - \mathrm{E}[a_{k+\ell}]\big)\,\right] \;=\; \lim_{k\to\infty} \mathrm{E}[a_k\, a_{k+\ell}] - \big(\mathrm{E}[a]\big)^2\,,$$

which measures the amount of correlation of a process with a time-shifted version of itself.

Finally, two processes a_k and b_k are *independent* if any subset of the variables $a_0, a_1, \ldots$ is independent of any subset of the variables $b_0, b_1, \ldots$.

Discrete-time processes with multidimensional state space

In the above, we have extended the notion of a random variable with values in V to that of a stochastic process with state space V. As with multivariate random variables, the state space of a stochastic process can be multidimensional as well. Specifically, let $V\!=\!\mathbb{N}^N$, then a discrete-time process with state space V is

$$\boldsymbol{a}_k \;=\; \{a_1, a_2, \ldots, a_N\}_k \;=\; \{a_{1,k}, a_{2,k}, \ldots, a_{N,k}\}\,. \tag{26}$$

As this notation suggests, the N-dimensional process $\boldsymbol{a}_k$ defines N 1-dimensional processes $a_{j,k}$, $1\leqslant j\leqslant N$. Bear in mind however, that these subprocesses are generally *not* independent of each other, so it is useful to keep them together. Note also that the state space of $\boldsymbol{a}_k$ is still countable, so we can call this process a *chain* as well.

1.2.4 Markov chains

A type of stochastic process that is of special importance in stochastic modelling is the Markov process. Consider a discrete-time chain a_k, i.e. the sequence $a_0, a_1, a_2, \ldots$ of random variables at time steps $k\!=\!0$, 1, 2, ... respectively. For a general process, the statistics of this process may be arbitrarily complex, i.e. the nth order characterisations (23) can have an entirely different structure for every other set of time indexes $(k_1, \ldots, k_n)$. Suppose we are following the realisation of such a sequence over time, then in principle, a variable a_{h+1} in the sequence may depend on everyone of the previous variables a_j, $0\leqslant j\leqslant h$. So given the observed values (the realisation) of the variables before step h+1, the distribution of the variable a_{h+1} at the next step is given by the conditional mass function

$$p_{a_{h+1}|a_h a_{h-1} a_{h-2}\ldots}(i|j_h, j_{h-1}, j_{h-2}, \ldots)\,,$$

[7]We do not consider *cyclostationary* processes for which (24) holds only for ℓ equal to multiples of a certain integer period $\delta > 1$.

where in general, all variables in the 'history' before step $h+1$ have influence on this distribution. However, if the structure of the stochastic process a_k is such that the conditional distribution of a_{h+1} depends on the value of the previous variable a_h only and is *independent of all previous values*, we say that the process has the *Markov property* and call it a *Markov chain*. More precisely, a process a_k with state space $\mathbb{N}$ is called a (onedimensional) Discrete-time Markov Chain (DTMC) if it satisfies the condition

$$\text{Prob}[a_{h+1}=j|a_h=i_h, a_{h-1}=i_{h-1}, \ldots] \;=\; \text{Prob}[a_{h+1}=j|a_h=i_h]\,, \tag{27}$$

for all time indices h and whenever the conditioning event has nonzero probability. As before, the set of possible values $\mathcal{S}$ of the variables in the sequence is the *state space* of the Markov chain, which we assume to be a subset of $\mathbb{N}$. Obviously, a Markov chain is much easier to describe statistically because of (27), since the state of the process at any given time contains all the information about the past evolution that is useful for predicting its future behaviour. If the Markov chain is stationary, or *time homogeneous* as this is also called, then (27) does not depend on h. For such Markov chains we can define the *transition probabilities* from state i to state j as

$$q_{ij} \triangleq \text{Prob}[a_{h+1}=j|a_h=i] \quad \text{for all } i, j \in \mathcal{S}\,,$$

for which clearly, $\sum_{j\in\mathcal{S}} q_{ij} = 1$ for any $i \in \mathcal{S}$. We shall only consider stationary DTMCs, which are fully described by

(i) the transition probabilities q_{ij} and

(ii) the *initial distribution*, i.e. the distribution of a_0, say $p_{a_0}(i)$, $i\in\mathcal{S}$.

Consider now a DTMC with finite state space $\mathcal{S} = \{1, 2, \ldots, M\}$. It is often advantageous to represent the parameters of the chain in matrix form as follows. The transition probabilities can be arranged in a matrix $\boldsymbol{Q}$ such that $[\boldsymbol{Q}]_{ij} = q_{ij}$,

$$\boldsymbol{Q} \triangleq \begin{bmatrix} q_{11} & q_{12} & q_{13} & \cdots & q_{1M} \\ q_{21} & q_{22} & q_{23} & \cdots & q_{2M} \\ q_{31} & q_{32} & q_{33} & \cdots & q_{3M} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ q_{M1} & q_{M2} & q_{M3} & \cdots & q_{MM} \end{bmatrix}.$$

This matrix $\boldsymbol{Q}$ is the *transition matrix* of the DTMC and has nonnegative elements with row sums equal to 1. Such a matrix $\boldsymbol{Q}$ is called a *stochastic* matrix or a Markov matrix and has many interesting properties, see Chapters 3, 4 and a.o. [30, 35, 82, 94, 141]. Let also the vector $\mathbf{p}_h$ represent the vector with the probabilities $\text{Prob}[a_h=i]$ as its elements, i.e. the pmf of a_h,

$$\mathbf{p}_h \triangleq \begin{bmatrix} p_{a_h}(1) & p_{a_h}(2) & \cdots & p_{a_h}(M) \end{bmatrix}\,, \tag{28}$$

then the initial probability distribution is given by the vector $\mathbf{p}_0$. It now follows from the Markov property (27) that

$$\mathbf{p}_h \;=\; \mathbf{p}_0\, \boldsymbol{Q}^h\,, \qquad h \geqslant 0\,, \tag{29}$$

which establishes the unconditional distribution of all variables a_h in the Markov chain using only the parameters $\boldsymbol{Q}$ and $\mathbf{p}_0$. Note that $\boldsymbol{Q}^h$ is itself also a stochastic matrix and its elements are sometimes called the *h-step* transition probabilities which we denote as

$$q_{ij}^{(h)} \;=\; \text{Prob}[a_{k+h}=j|a_k=i]\,.$$

Note that all of the above considerations are still valid if $\mathcal{S}$ is countably infinite, i.e. if the chain has $M=\infty$ different states.

Classification of the states

Given a Markov chain a_k, its states in the set $\mathcal{S}=\{1,2,\ldots,M\}$, which is either finite or countably infinite, can be classified in a number of ways. The following characterisations of the states in $\mathcal{S}$ are all based on the transition probabilities q_{ij} and are important when considering the asymptotic behaviour of the Markov chain, i.e. the distribution $\mathbf{p}_h$ for $h\rightarrow\infty$.

- Let $i,j\in\mathcal{S}$ be two states of the Markov chain, then state j is said to be *reachable* from state i if

  $$\exists h>0\;:\;q_{ij}^{(h)}>0\,,$$

  i.e. if starting from state i, there is a nonzero probability that after one or more steps the chain ends up in state j . For convenience, let us call $A(i)\subset\mathcal{S}$ the set of all states that are reachable from state i. If $A(i)=\{i\}$ or equivalently, $q_{ii}=1$ then i is called an *absorbing* state. Once the chain enters an absorbing state, it will remain there forever. Also, two states $i,j\in\mathcal{S}$ *communicate* with each other if i is reachable from j and vice versa.
- We say that a state $i\in\mathcal{S}$ is *recurrent* if for every j that can be reached from i, we have that i can also be reached from j. In other words, if $j\in A(i)$ then also $i\in A(j)$. If a state i is not recurrent, then it is *transient*. In particular, this means that there are states $j\in A(i)$ such that $i\notin A(j)$. The chain is sure to enter such a state j sooner or later, after which it can never visit state i again. Another way of determining the recurrent or transient character is to consider the *recurrence time* τ_i for state i, defined as the random variable

  $$\tau_i\triangleq\min\{h>0 : a_{k+h}=i \text{ given that } a_k=i\}\,.$$

  So if $\tau_i=h$, then the first return to state i occurs h steps after leaving state i. The state i is transient if τ_i is a deficient random variable, i.e. if $\text{Prob}[\tau_i=\infty]>0$. On the other hand, if $\text{Prob}[\tau_i=\infty]=0$ then the chain will always return to state i.
- For a recurrent state $i\in\mathcal{S}$, we can make a further discrimination based on the expected value of the recurrence time τ_i. Let us define

  $$\mu_i\triangleq\text{E}[\tau_i]\,,\qquad i\in\mathcal{S}\,,$$

  then the state i is called *positive recurrent* if $\mu_i<\infty$ and *null recurrent* if $\mu_i=\infty$. In the latter case, the chain does indeed return to state i, but on average takes an infinite number of steps to do so. Note that this can only occur if $\mathcal{S}$ is countably infinite. A DTMC with finite state space has only transient and positive recurrent states.

► Additionally, if state $i \in \mathcal{S}$ is recurrent, we can define its *period* δ_i as

$$\delta_i \triangleq \gcd\{h > 0 : q_{ii}^{(h)} > 0\}\,.$$

A recurrent state i is called *aperiodic* if $\delta_i = 1$, and *periodic* if $\delta_i > 1$.
The behaviour of a DTMC is determined by the classification of its states according to these definitions. The chain a_k with state space $\mathcal{S}$ is called *irreducible* if all states in $\mathcal{S}$ communicate with each other. It is clear that an irreducible DTMC can have no transient states. But even if we leave out all transient states, the recurrent states of a general DTMC do not necessarily all communicate with each other. There may be several subsets in $\mathcal{S}$ of communicating states which are called *recurrence classes*. So all states of an irreducible DTMC are in the same recurrence class.

It can be shown [82] that the states of an irreducible DTMC are all of the same type according to the definitions above. In other words, all states are either null recurrent or positive recurrent. Furthermore, all states have the same period $\delta = \delta_i$ for all $i \in \mathcal{S}$. Hence, we call an irreducible DTMC periodic or aperiodic, and null or positive recurrent if at least one of its states can be classified as such. In case the DTMC is periodic with period $\delta > 1$, the states can be partitioned in δ sets $\mathcal{S}_0, \mathcal{S}_1, \ldots, \mathcal{S}_{\delta-1}$ such that if the chain is in $\mathcal{S}_d$ at time k, it will be in $\mathcal{S}_{(d+1) \bmod \delta}$ at time $k+1$.

Ergodicity

In general, a stochastic process is *ergodic* if all its statistical properties over the ensemble can be determined from time averages over one realisation only. The usefulness of this property for us lies in the fact that the converse is also true. If we can calculate the (stationary) distribution of a certain process over the ensemble and this process is ergodic, then we are sure that every realisation of this process will have the same properties when observed long enough over time[8].

For Markov chains in particular, it can be shown that if a DTMC is irreducible, aperiodic and positive recurrent, then it is also ergodic and vice versa. However, before discussing the implications of ergodicity, let us first have a look at the long-term behaviour of a DTMC a_k with transition matrix $\boldsymbol{Q}$. We call a *stationary* distribution $\mathbf{p}$ of the DTMC a_k a probability vector as in (28) over the states of $\mathcal{S}$ such that a one-step transition has no effect on these state probabilities. Clearly, from (29) it then follows that

$$\mathbf{p} = \mathbf{p}\,\mathbf{Q}\,, \qquad \text{and} \qquad \mathbf{p}\,\mathbf{1} = 1\,, \tag{30}$$

where $\mathbf{1}$ is a column vector with every element equal to 1. For a general DTMC without null recurrent classes, a stationary distribution always exists, although there may be more than one.

[8]Note that the whole idea of analytic stochastic modelling as a means of performance prediction relies heavily on the assumption of ergodicity. The results from an analysis such as those given in further chapters, are expressions for (part of) the statistical properties of the stochastic process that describes a certain system or model. These statistics are derived from the whole ensemble, i.e. from the set of *all* possible realisations of the stochastic process. However, the motivation for doing the analysis in the first place, is often to predict the evolution of only *one* realisation of the process over a long period of time. So the desired measures and those derived from the analysis are in fact two different things, which are equal only if the process in question is ergodic. The ergodicity requirement is tacitly assumed.

In contrast, a *limiting* distribution does not exist for all DTMCs. Suppose that the initial distribution is given by $\mathbf{p}_0$, then it follows from (29) that after a very large number of steps, the state distribution of the DTMC becomes

$$\mathbf{p}_\infty = \lim_{h\to\infty} \mathbf{p}_h = \mathbf{p}_0 \lim_{h\to\infty} \boldsymbol{Q}^h = \mathbf{p}_0 \boldsymbol{Q}^\infty , \tag{31}$$

at least if this limit exists. Obviously, if $\mathbf{p}_\infty$ exists, then $\boldsymbol{Q}^\infty$ exists as well and is stochastic. For a general DTMC, the existence of a limiting distribution $\mathbf{p}_\infty$ and the values of its elements depend on the initial distribution $\mathbf{p}_0$ in (31). For example, if $\mathbf{p}_0$ only contains nonzero probabilities for states within a certain number of recurrence classes, then the chain can never reach the other recurrence classes, so the probabilities of $\mathbf{p}_\infty$ (if it exists) will be zero for the states in the other classes as well. On the other hand, whatever the starting probability $p_{a_0}(i)$ in a transient state i, we always have that $p_{a_\infty}(i)=0$. So in general, a DTMC can exhibit many different kinds of long-term behaviour, depending on the starting conditions.

We are particularly interested in the conditions under which a DTMC has an existing limiting distribution $\mathbf{p}_\infty$ (and thus also an existing $\boldsymbol{Q}^\infty$) that is also independent of the initial distribution $\mathbf{p}_0$. Clearly, for such a DTMC, the limiting distribution $\mathbf{p}_\infty$ is *unique* and equal to the stationary distribution $\mathbf{p}$, since both must satisfy (30). Moreover, the fact that $\mathbf{p}_\infty$ is independent of $\mathbf{p}_0$ in (31), implies that all rows in $\boldsymbol{Q}^\infty$ are equal to the limiting vector $\mathbf{p}_\infty$. The conditions for a unique $\mathbf{p}_\infty$ to exist are the following:

- ▶ As we have seen, if the DTMC contains multiple recurrent classes, the limiting distribution will depend on the initial state distribution $\mathbf{p}_0$, so we only allow one recurrence class. Transient states can be ruled out as well, since they do not affect the limiting distribution. Therefore, we only consider DTMCs which are *irreducible*.
- ▶ Clearly, if the (irreducible) DTMC is periodic, the limit (31) cannot exist. If at time k, the chain is in a state $i\in\mathcal{S}_d$, $d=0,\ldots,\delta-1$, then the probability mass will be concentrated exclusively in the states of $\mathcal{S}_{(d+1) \bmod \delta}$ at time $k+1$, in $\mathcal{S}_{(d+2) \bmod \delta}$ at time $k+2$, and finally back in $\mathcal{S}_d$ at time $k+\delta$. The cycling between the different state partitions goes on forever and thus the limit in (31) never converges. So in order for a limiting distribution to exist, the DTMC must be *aperiodic*.
- ▶ In case $\mathcal{S}$ is finite, the above conditions (irreducible and aperiodic) are enough to ascertain a unique limiting distribution. However, if $\mathcal{S}$ is countably infinite, the states must also be *positive recurrent* in order for $\mathbf{p}_\infty$ to exist independently of $\mathbf{p}_0$.

As we have already mentioned, a DTMC satisfying these requirements[9] is *ergodic*. More specific, a DTMC is ergodic if and only if it has a unique limiting distribution $\mathbf{p}_\infty$, which we then call $\boldsymbol{\pi}$,

$$\boldsymbol{\pi} \triangleq \begin{bmatrix}\pi_1 & \pi_2 & \ldots & \pi_M\end{bmatrix} = \lim_{h\to\infty} \begin{bmatrix}p_{a_h}(1) & p_{a_h}(2) & \ldots & p_{a_h}(M)\end{bmatrix} .$$

Roughly speaking, we can say that the aperiodic and positive recurrent property assure that $\boldsymbol{\pi}$ exists, whereas irreducibility is required for $\boldsymbol{\pi}$ to be unique and independent of $\mathbf{p}_0$.

[9] These conditions are not always easy to check in practice. However, some *sufficient* conditions for ergodicity have been given by Foster [88, 158].

In the remainder of this thesis, we will assume that all DTMCs we encounter are ergodic[10] and refer to the limiting distribution $\boldsymbol{\pi}$ as the *equilibrium distribution*, which is thus determined by

$$\boldsymbol{\pi} = \boldsymbol{\pi}\,\mathbf{Q}\,, \qquad \text{and} \qquad \boldsymbol{\pi}\,\mathbf{1} = 1\,.$$

An interesting consequence of ergodicity is the fact that $\pi_i > 0$ for all states $i \in \mathcal{S}$. Moreover, every state i will be visited an infinite number of times and the time between the visits has finite mean $\mu_i < \infty$. Recall that for an ergodic process, the statistical properties over the ensemble can be obtained as a suitable time average over one realisation. One way to apply this, is the observation that the proportion of time spent in state i during n consecutive steps converges to π_i for $n \to \infty$:

$$\lim_{n\to\infty} \frac{\#\{k : a_k = i, 0 \leqslant k < n\}}{n} = \pi_i\,, \qquad \forall i \in \mathcal{S}\,,$$

which provides an alternative way to calculate $\boldsymbol{\pi}$. Equivalently, averaging the recurrence times τ_i, we find that

$$\pi_i = \frac{1}{\mathrm{E}[\tau_i]} = \frac{1}{\mu_i}\,, \qquad \forall i \in \mathcal{S}\,.$$

Markov chains with multidimensional state space

There is no reason why the states in the space $\mathcal{S}$ should not be multivariate. In fact, the methodology in this thesis heavily relies on DTMCs with *multidimensional* state space. Consider a discrete-time chain $\boldsymbol{a}_k$ as in (26) with state space $\mathcal{S} = \mathbb{N}^N$, and denote a particular state as $\boldsymbol{i} = \{i_1, i_2, \ldots, i_N\}$, then the Markov property (27) for a multidimensional chain $\boldsymbol{a}_k$ can be stated as

$$\mathrm{Prob}[\boldsymbol{a}_{h+1} = \boldsymbol{j} | \boldsymbol{a}_h = \boldsymbol{i}, \boldsymbol{a}_{h-1} = \boldsymbol{i}', \ldots] = \mathrm{Prob}[\boldsymbol{a}_{h+1} = \boldsymbol{j} | \boldsymbol{a}_h = \boldsymbol{i}]\,.$$

As $\mathcal{S}$ is still countable[11], it is clear that all discussed properties of DTMCs with univariate state space are equally valid in case of a multivariate state space. In particular, the classification of the states $\boldsymbol{i} \in \mathcal{S}$, as well as the requirements for ergodicity remain unchanged. It is also important to note that the subprocesses $a_{n,k}$, $n = 1, \ldots, N$ by themselves are generally no Markov chains.

1.3 Stochastic modelling of queueing systems

1.3.1 Modelling

In sciences, a commonly used method to gain insight into complex phenomena is *modelling*, i.e. replacing the realistic object under study by a simplified abstract *model*. We

[10] In fact, we will never talk of ergodicity in explicit terms again. Rather, we say that a system is 'stable' or 'reaches stochastic equilibrium', by which we mean that the DTMC describing this system is ergodic.

[11] A set of multivariate states can always be rewritten as a set of univariate states by suitably relabelling the states. So mathematically, there is no real difference between a one-dimensional and a multidimensional state space as long as both are countable. Nevertheless, it may be difficult to interpret the relabelled representation of the state space, so it is usually better to retain the multivariate representation.

too, when studying the performance of communication systems, rely on this paradigm to make predictions about the behaviour of realistic systems.

We avoid the philosophical discussion on the question what a system actually is, although we can assume that every dynamical phenomenon can be seen as a system on one level or another. Nevertheless, with regard to telecommunications, we can identify a *realistic system* to be every device, or part of a device, in which information is manipulated. So a communication system can be either conceptual or tangible, can be implemented in software, hardware or both. Obviously, these realistic systems can be extensive and very complex (like e.g. a layered computer network) with relation to the actions they perform on the manipulated information. To make matters worse, their dynamic behaviour may depend on a multitude of factors that interact in complex ways. So often, the dynamic operation of a realistic system is too complex to be captured entirely in the mind of an outside observer. Fortunately however, we are usually not interested in encompassing and describing every subtle detail of a system, but only in certain aspects of it that are useful to us. In answer to the question why a system behaves as it does on a certain level, why it exhibits a specific observable way of operating, we must identify and focus on the particular underlying rules of interaction that most prominently cause the observed behaviour on that level. It is therefore left to the observer's own judgement to find out how deep the rabbit hole goes, i.e. to decide how much detail is required and to concentrate only on those aspects that are relevant.

In order to study a realistic system, to be able to understand it and make sensible assertions or predictions, we must rely on a *model* of that system. A model can generally be seen as an abstract, simplifying reflection of a realistic system, that is easy to handle and particularly focusses on aspects that are of interest. It is a tool that enables us to answer questions about the real system to which the model corresponds. As such, after the observer has identified those aspects of the system that are relevant to his purposes, he can translate these mechanisms into an abstract model. In doing so, the multitude of complex interactions at the various levels of detail in the real system are aggregated, formalised, reduced, simplified or very often simply ignored, such that only the most prominent dynamic properties remain, in the form of simple logical or mathematical operations. Again, *which* aspects of the system should be retained in the model and on what level they should be described, depends on the specific question that the observer wants to answer and how: do we need qualitative or quantitative answers, should they be very precise or only rough estimates? So we can either choose a simple model accounting for only a few aspects of the system, or an elaborate, complex model that is very detailed and accounts for a lot of relevant effects in the system. In the former case, the results will be crude but easy to obtain, whereas in the latter, the results may be very precise but are generally much harder to arrive at. Therefore, a trade-off between accuracy and tractability always has to be made when choosing an appropriate model.

In this thesis, we specifically focus on those aspects of telecommunication systems that involve the *queueing* of fixed-size information packets as the distinguishing relevant feature. For such systems, it is common practice to use *stochastic models*. These models are event-based and capture the dynamics of a system with two kinds of *rules*:

- *Deterministic rules*
  Usually the deterministic rules reflect the core aspects of the system, i.e. how a certain protocol or queue is *designed* to operate. For example, the employed *discipline* (see further) for queueing systems is usually a set of deterministic rules, indicating precisely what happens if a packet arrives or leaves. In view of their central role, we may regard the deterministic rules as 'internal' to the system.
- *Stochastic rules*
  On the other hand, a stochastic description is used whenever the underlying causes are unknown or too complex. For example, again for queueing systems, 'external' factors such as the time instants on which packets arrive to the queue, or the number of packets contained in an arriving message are modelled as stochastic rules. In e.g. the latter case, we then represent the number of packets as a stochastic variable ℓ with a certain pmf $p_\ell(n) = \text{Prob}[\ell = n]$, $n \geqslant 1$. As such, the model does not concern itself with computing the value that ℓ would have in the real system, but rather *assumes* the mathematical construct of a random variable. In this way, the model 'cuts away' the difficulty of having to account for the many, hopelessly complex interacting circumstances that lead to the value of ℓ in the real system. Obviously, the pmf $p_\ell(n)$ in the model must be chosen to represent the message sizes occurring in the system as closely as possible.
  So a stochastic description is useful for modelling those aspects that are out of the scope of interest, or not part of a design that needs to be evaluated[12], but are nevertheless important for an adequate prediction of the system's behaviour. If the model allows for *generally* distributed random variables to be used, then their distributions are *input* parameters to the model.

Somewhat confusingly, the terms 'system' and 'model' are used interchangeably in this thesis. From this point on, if we refer to a 'system' it will be clear from the context whether we mean a realistic device or phenomenon, or rather the simplified, mathematical abstraction that is the system's *model*. In fact, although the models we treat are *motivated* from realistic systems, no effort is made to compare the results obtained with the models to measurements of real systems. Also, the emphasis of this thesis is foremost on the techniques used to analyse the models rather than on the applicability to real situations. So real systems are not considered explicitly here, and we can thus reserve the term 'system' to mean an abstract model in the sense of the above discussion, without much ambiguity. Hence, in what follows, a 'queueing system' is usually the same as a 'queueing model'.

1.3.2 The choice for discrete time

In the literature on queueing theory, two main classes of models can be discerned, differing with regard to how *time* is modelled. Either the model assumes time to be continuous, in which case events can occur on all possible instants, or the model assumes that changes in the system can only occur at discrete points in time. These are called *continuous-time* and *discrete-time* models respectively and both classes have different sets of analytic solution techniques and typical methodologies.

[12]Some queueing systems are stochastic in design as well, and use a so-called *stochastic* discipline or protocol.

As of the days of Erlang, until about 1960, it was self-evident to model time as a continuous dimension, because of the analog operation of most communications systems in that era. For example, in a telephone switching center, a call can come in at any time instant and have any duration. However, nowadays most communication systems work digitally and on much smaller timescales. For such systems, it is useful to construct a model in discrete time, since both space and time in these systems are often *already* discrete dimensions. For example, in the CPU of a digital computer, only one instruction can be performed per clock cycle, so the state of the system cannot change in between ticks of the clock. Also, in the lower layers of a digital communication network, recall that we have assumed all packets to be equal in size, so it is logical to assume that the time required to process such a packet is constant as well. Often, packets are small and can be processed fast. Hence, while the system is busy with one packet, it can do little else and no significant changes occur until the packet in question is finished.

So for a digital system, a suitable timescale can usually be found on which the realistic system is already *synchronised*. In a discrete-time model for such a system, we therefore assume the same synchronisation and divide time (which is continuous in principle) in fixed length *slots*, corresponding to this timescale. The modelled system is then synchronised on these slots, which means that changes can only occur at slot boundaries. A slot can then be a clock cycle, or a specific parameter of some communication protocol. However, we will always take a slot to be the time required to process one packet. Because of this dimensional relation between slots and packets, our models are not only discrete in time, but also discrete in space. Indeed, every quantity that characterises the system is either an integer number of slots or an integer number of packets, which explains why we were only interested in discrete random variables and processes in Sec. 1.2. Some of the first publications that analyse discrete-time queues under these assumptions are [133, 139].

Nevertheless, the use of discrete-time models is not necessarily limited to systems with a 'natural' synchronisation. Also asynchroneous systems can be described adequately in discrete-time, which allows the techniques from the discrete-time realm to be used in those cases as well. As we demonstrate in Fig. 1.4, the only concern is to take the slots as small as possible. At the top, we show the evolution over time of a continous-time (CT) queueing system with asynchroneous packet arrivals, where each packet requires exactly one second to process. The curves below indicate the evolution of the queue content in corresponding discrete-time (DT) systems for a decreasing slot length of 1s, 0.5s, 0.25s and 0.1s. It is seen that the DT systems converge to the CT system for smaller slot lengths, as can be expected. It is important to note that for the DT systems, the arriving packets are not stored in the queue before the end of the slot in which they arrived. Obviously, this is because changes are only allowed on slot boundaries for DT systems. This assumption holds throughout this thesis.

1.3.3 Characteristics of a queueing system

A typical single-server *queueing model* to which all models treated in this thesis comply, is shown in Fig. 1.5. We assume the *system* or *queue* to be a structure comprising the following two objects:

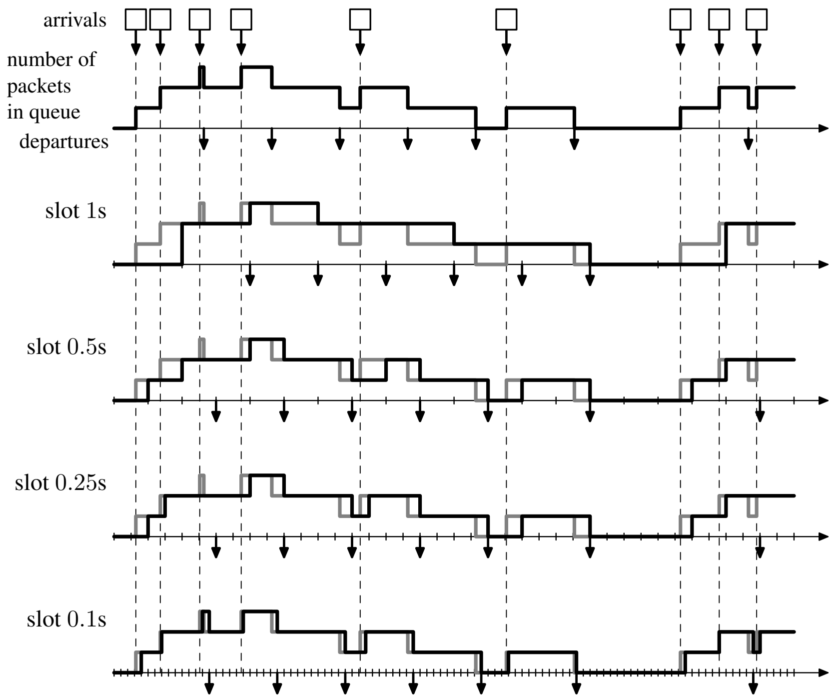


Figure 1.4: First, the continuous-time system is shown, with the packet arrivals, departures and the bold line indicating the number of packets in the queue. The continuous-time system can be approximated by a discrete-time system with increasing precision as the slot length is chosen smaller and smaller.

- *A server*
  All packets that enter the queue eventually want to receive some *service*, which is the abstraction of whatever manipulation of the information in the packets is required in the communication system. The 'server' is the place where exactly one packet at a time can be served. We only consider models with *one* server.
- *A buffer*
  The goal of the buffer is to provide a temporary storage facility for arriving packets that cannot be served immediately upon arrival. For instance, if multiple packets arrive in an empty system during a slot, only one of them can receive service during the next slot and the others have to wait. The number of available spaces C in the queue, i.e. the maximum number of packets that can be stored is called the *capacity* of the queue. We always adopt the theoretical assumption of an *infinite*-capacity queue, $C = \infty$, mainly because this allows an efficient analysis of the system in the transformation domain.

The evolution of the system is further determined by the specific modelling assumption made for each of the following parts of the stochastic queueing model:

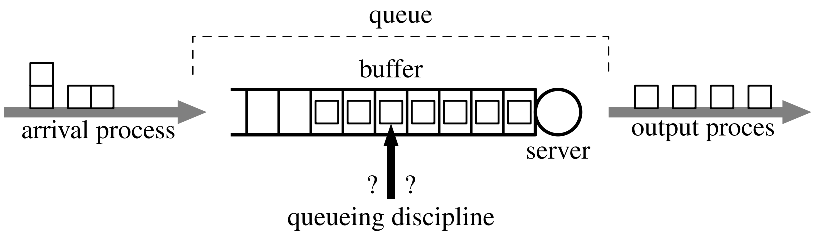


Figure 1.5: Characteristics of a typical queueing system with one queue and one server.

- *Arrival process*
  The arrival process is the result of modelling the conditions in an upper network layer. As we have discussed, these conditions are generally very complex, so a *stochastic* description is in order. For DT systems, the numbers of packet arrivals in each of the subsequent slots form a discrete-time chain that is an 'input' parameter to the model. In chapters 3 and 4 for instance, the number of packet arrivals a_k forms an independent process, i.e. the variables $a_0, a_1, a_2, \ldots$ are independent and identically distributed (iid) with common pgf $A(z)$. In chapter 2 on the other hand, the sequence a_k exhibits a particular correlation structure, of which we study the impact on the queueing behaviour. In CT systems, the arrival process is usually characterised by the inter-arrival times between individual packet arrivals.
- *Queueing discipline*
  The queueing discipline or *scheduling mechanism* of the system is a set of (mainly *deterministic*) rules that are used to decide which packet in the buffer will be chosen for service whenever the server becomes available. There are many different useful disciplines, each designed to make the queue behave in a particular desired way. The most well known discipline is probably FIFO (First-In First-Out), which selects packets in the order of their arrival to the queue. LIFO (Last-In First-Out) on the other hand always selects the latest arrival, whereas ROS (Random Order of Service) [122] chooses a packet at random. Some disciplines are designed to distinguish between two or more types of packets, each with different QoS requirements. For example, AP (Absolute Priority) [196, 197] selects an available packet of the highest priority. Other such disciplines are e.g. RR (Round Robin), WFQ (Weighted Fair Queueing) [69] and their variants like EDF (Earliest Deadline First) [140, 175]. Some disciplines like FIFO, LIFO, AP, RR, WFQ and EDF are purely deterministic, whereas others like ROS use stochastic rules as well. Also, it is clear that not all disciplines maintain the order of packet arrivals. In chapter 4, we present a new queueing discipline called the Reservation discipline that is intended to provide a better QoS to one type of packets in a controllable way.
- *Service process*
  In the same way that the arrival process models the upper layer, the service process models the conditions in the network layer below. So here too, a *stochastic* de-

scription is preferred. A first relevant parameter is the number of servers c and their availability throughout time. As already indicated, we only consider models with a single server that is busy serving a packet whenever one is available in the system. Secondly, the distribution of the *service times* must be given as well, i.e. the time required to serve one packet. For DT systems, the service times of the consecutive packets form a discrete-time chain of which the statistical properties must be specified. In chapters 2 and 4, the service time of each packet is exactly one slot. In chapter 3 on the other hand, the 'service times' of the transmitted packets turn out to be a sequence of nonindependent random variables.

The main goal of a queueing model as described here, is to provide results (analytic or otherwise) with regard to the distribution of certain *performance measures* of interest. In this thesis, we mainly focus on the equilibrium distribution[13] of the following measures:

► *Queue content*
The number of packets present in the system during a certain slot k is called the *queue content* or *system content* and is denoted by u_k. The sequence $u_0, u_1, u_2, \ldots$ forms a discrete-time chain, but usually not a Markov chain. By u, we denote a random variable of which the distribution is the equilibrium distribution of the sequence u_k. This distribution follows from our analysis as the pgf $U(z)$ obtained in terms of the model parameters. This pgf tells us exactly what the probability is of finding a certain number of packets in the queue during an arbitrary slot under equilibrium conditions. Obviously, this is important information when dimensioning the capacity of the queue in a realistic system.

► *Delay*
The delay of a packet is defined as the total number of slots it spends in the system, from the moment it is stored in the queue, until the moment it departs. Under equilibrium conditions, we denote the delay of an arbitrary packet as d, of which the pgf $D(z)$ will also be calculated from the parameters of the model. This delay distribution can be useful for dimensioning as well. For instance, if the delay is too high, appropriate actions must be taken, such as decreasing the mean arrival rate or decreasing the service times. In case of the model in chapter 2, each packet belongs to a message and we also consider the delay, the *waiting time* and the *transmission time* of messages, to be defined precisely later on.

Another performance measure of practical importance is the *packet loss ratio* (PLR), i.e. the fraction of arriving packets that are not admitted to the system because there is no more room in the queue. Obviously, this is only of concern in finite-capacity queues since packets need never be rejected under the assumption of infinite capacity $C = \infty$ made above. Fortunately, the PLR of a system with a capacity C that is finite but large, can be estimated rather accurately from the tail distribution of the queue content u in the corresponding system with $C = \infty$.

[13] As discussed earlier, the importance of the equilibrium distribution of an ergodic system lies in the fact that the long-term behaviour of one realisation can be derived from it. Nevertheless, some authors are also interested in the transient distributions of these measures, starting from a certain initial state of the system, see e.g. [40, 56, 198].

1.3.4 The Discrete Supplementary Variable Technique, DSVT

In each of the following chapters, a different queueing model is presented and analysed in full detail. The basic methodology of queueing analysis however, is the *same* for each of the three models we focus on, at least in principle. The idea behind the method is to describe the system as a Markov chain with a multidimensional state space.

The typical analysis is carried out in the transformation domain and the subsequent steps in such an analysis can be summarised as follows:

- Set up the stochastic model. Identify and suitably represent the relevant quantities in the system as random variables.
- Quantify precisely the interactions between these variables and their evolution from one slot to the next. The characterisation of this evolution can usually be expressed as a set of *system equations*. This step requires insight in the 'physical' properties of the system. Every possible situation has to be taken into account.
- Identify a suitable description of the system's state during an arbitrary slot k which contains enough information so that the intended performance measures can be derived from it. In practice, this is usually a subset of the random variables identified in the first step. The variables that are chosen to be part of the system state are called the *system state variables* and we characterise their (joint) distribution in slot k by their joint pgf P_k. The primary concern here, is to choose the collection of system state variables in such a way that it is Markovian, in the sense that P_{k+1} can be determined from P_k without relying on the system state variables in the slots before slot k. Note that a 'natural' candidate to be part of the system state is the variable u_k, i.e. the queue content.
- As we have said, the system equations relate the system state in slot $k+1$ to the system state in slot k. In this step, we want to translate this relation to the transformation domain. Specifically, establish the relationship between the joint pgf P_{k+1} of the system state variables in slot $k+1$ to the corresponding joint pgf in slot k by using the information in the system equations. Mathematically, this usually is the most involving part of the procedure.
- Now assume that for $k \to \infty$, the system reaches stochastic equilibrium such that the distributions P_k and P_{k+1} converge to the same limiting pgf P. The obtained relation between P_k and P_{k+1} then results in a *functional equation* that determines the equilibrium distribution P in an implicit way.
- Sometimes, the functional equation for P can be solved explicitly, in which case the joint distribution of the system state variables can be obtained in closed form. However, even if an explicit solution for P is not available (at least not in a form that is useful), it is still possible in most cases to obtain relevant characteristics of the marginal distributions of the system state, such as e.g. the moments or approximated tail distributions. For instance, the queue content u_k is usually one of the system state variables. Hence, the moments $\mathrm{E}[u], \mathrm{E}[u^2], \ldots$ can be obtained explicitly from the functional equation, even if P cannot be determined explicitly. We refer to chapter 2 for a demonstration.
- The equilibrium distribution P (or the moments thereof) forms the starting point of the further analysis. The distribution (or the moments) of some performance

measures that are directly or indirectly related to the variables in the system state can then be inferred. For example, the packet delay is usually not suited to be part of any system state[14], but its distribution $D(z)$ can nevertheless be calculated from P.

Historically, this method has been called the 'method of the supplementary variable' [62, 120] and has been successfully applied for CT and DT systems, both in the probability domain and the transformation domain. It is called this way, because supplementary variables are added to the system state in order to make it satisfy the Markovian property. For convenience, we call the specific version of this method used here the *Discrete Supplementary Variable Technique* (DSVT), in which the system is described as a multidimensional DTMC and the analysis is performed mainly by using pgfs. Although not their invention, the DSVT has been much refined and applied to many different queueing systems in the works of Bruneel, see e.g. [43].

The question remains how many and which variables should be included in the system state. The answer to this question is not always easy, since multiple considerations are of importance. In the first place, the system state should always be Markovian. Secondly, the choice also depends on the specific performance measures that we want to obtain with the analysis. For example, if only the equilibrium distribution of the queue content u is desired, then sometimes significantly *less* variables are required than if the packet delay d must be calculated as well.

We have seen the conditions under which a general DTMC is ergodic and thus reaches equilibrium in Sec. 1.2.4. In our summary of the DSVT, we have simply *assumed* that the multidimensional Markov chain reaches stochastic equilibrium for $k \to \infty$ if some straightforward stability condition is met. However, this condition usually follows from intuitive reasoning rather than mathematically ascertaining the ergodicity of the chain (i.e. checking irreducibility, aperiodicity and null recurrency). Although not mathematically rigorous, it is safe to assume ergodicity from a practical point of view if there is no direct evidence in the physics of the queueing model that would indicate otherwise. In our experience, periodic, nonstationary or deterministic behaviour of e.g. the queue content, can only occur if one or more of the stochastic 'input' processes are periodic, nonstationary or have some particular deterministic pattern. So the system simply *inherits* these undesired properties from the input processes (e.g. the arrival process, server availability process or stochastic scheduling rules) but does in general not *produce* them[15], unless specifically introduced by the deterministic system rules (the queueing discipline). In any case, if stochastic equilibrium is not reached, there will be a clear and identifiable reason for that, found either in the system's queueing discipline or in one of the stochastic input processes.

[14]The packet delay is defined with relation to a *packet*, not with relation to a *slot* as is the case for the system state variables. In other words, contrary to e.g. the queue content, it is senseless to talk about the 'packet delay during a certain slot'.

[15]I.e. 'garbage in, garbage out'.

Chapter 2

A multiplexer with correlated train arrivals

At the edge of an ATM (Asynchronous Transfer Mode) network, large external data frames, e.g. IP (Internet Protocol) frames, are segmented into fixed-length ATM cells, which are then transported through the network [103, 106]. In order to model the correlation in traffic streams that results from this segmentation, several researchers have analyzed discrete-time queueing systems where the customers are messages consisting of a random number of fixed-length packets; see, e.g. [45, 47, 54, 57, 63, 67, 207, 208, 211, 219]. Here, the term *message* corresponds to the external data format, whereas the term *packet* is used to denote the internal format. The segmentation process of the messages into packets at the network layer is conceptually shown in Fig. 2.1. Time is assumed divided into fixed-length intervals, referred to as slots, such that one slot suffices to transmit exactly one packet from the buffer. A message enters the queue like a train at the rate of one packet per slot and packets are assumed to leave the buffer at the end of a slot. Various distributions for the lengths of the messages have been considered: a geometric distribution [45,47,208,219], constant-length messages [54,211] and an arbitrary distribution [67,207]. The aforementioned papers however all assume

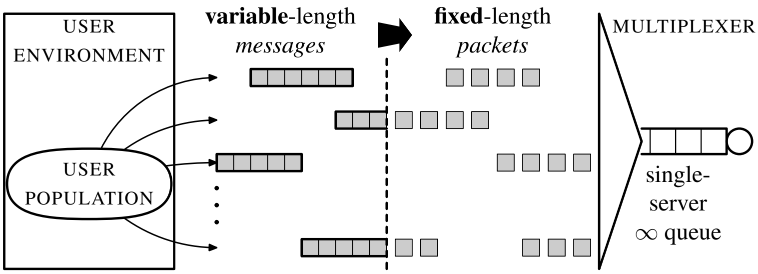


Figure 2.1: The messages generated by the user population are broken down to trains of fixed-size packets.

that the numbers of newly generated messages during the consecutive slots constitute a sequence of independent and identically distributed (iid) random variables. A somewhat related queueing model is considered in [57], where the delay of messages consisting of a fixed number of packets is treated under the assumption of an uncorrelated packet arrival process. In [63], a queueing model with multiple classes of messages is studied. Depending on its class, the packets of a message arrive either as a train or as a batch, with the one having priority over the other.

An important difference of our model with that of most of the existing studies about multiplexer models (see e.g. [45, 129, 194, 219]) is that we do not consider the arrival stream to be the superposition of traffic coming from a finite number of sources. Instead of characterising individual sources, we propose a *global description* of the aggregate packet arrival stream, as explained in the next section. Our model envisions a 'multiplexer' in a more generic sense than the classical device with a fixed number of limited-bandwidth input lines. In the classical sense, each message is sent to the queue by a 'user' claiming an input line with a link capacity of one packet per slot. The number of concurring messages (or active users) is then bounded by the number of input lines. Instead as an aggregate of individual sources, the arrival stream in our model comes from *one* centralised source that can be seen as an unbounded pool of 'users', capable of sending one message at a time, also at one packet per slot. There is no upper limit on the number of messages (users) that can simultaneously be active, so there is no bandwidth limitation other than the stability condition of the queue. Neither does the generation of new messages depend on how many of them are already active.

The analysis in this chapter is the result of an attempt to take into account a possible correlation in the message generation process. In particular, the distribution of the number of newly generated messages or leading packet arrivals in a slot is assumed to depend on the value of some environment variable, which represents the behaviour of the user population. There are two possible environment states, each with geometrically distributed sojourn times. In particular, our analysis yields results with regard to

- *Packets*:
  the mean, variance and tail distribution of the number of packets in the queue during equilibrium, as well as the mean delay of an arbitrary packet,
- *Messages*:
  the mean delay and mean transmission time of an arbitrary message during equilibrium.

Note that the considered *correlated packet-train arrival process* contains two types of correlation: a *primary* correlation, which results from the fact that one message may cause packet arrivals in the queue during several consecutive slots, and a *secondary* correlation, which is due to the nonindependent generation of new messages. It is important to note that these types of correlation are strictly Markovian by nature and have an exponential decay over larger time lags. Our arrival model thus qualifies as strictly short range dependent (SRD) and our analysis as it stands now is unable to cope with long range dependent (LRD) traffic.

2.1 The nonindependent generation of messages

We study a discrete-time single-server queueing system with infinite storage capacity. A user population generates entities, referred to as messages, which consist of a variable number of fixed-length packets. These messages are delivered to the buffer at a rate equal to the output link rate of the queue, i.e. one packet per slot (*train arrivals*). Note that we do not assume a limit on the number of users simultaneously sending packets to the multiplexer (i.e. being active). Hence, the number of packet arrivals per slot is not necessarily bounded.

In practice, our generic multiplexer model with its particular types of correlation in the arrival stream, is suited to describe queues at places where input bandwidth is not an issue (like inside a node or a processor) and where the origin of the burstiness is centralised (like traffic coming mainly from one application). This may for instance be the case in systems where larger blocks of information (messages) are at some point transformed into contiguous streams of smaller fixed-size packets. Suppose the multiplexer operates at the level of a packet-based network layer and its input originates from an application in the layer immediately above. When a message is generated by the application (a user population in the upper layer), it is presented to the lower layer for transmission. However, since the lower layer is packet-based, the message needs to be broken down in packets at the edge of the two layers. As is the case in many realistic systems, the packets originating from one transmission request (one message) do not enter the multiplexer as a batch, but as a train: one by one. It is obvious that, due to this origin, the aggregated packet stream in the lower layer will be heavily correlated (*primary* correlation): for instance, if a message with a length of two packets is generated during some slot, it is sure that at least one packet will arrive to the multiplexer during the next slot, whereas this doesn't need to be the case for any arbitrarily chosen slot.

To be able to analyse the multiplexer performance, we still need to specify the message generating behaviour of the user population. Instead of just assuming that this is an uncorrelated process, we want to assess the impact of a possible correlation or burstiness in the generation of messages at the higher layer. Therefore, we extend the model of [207] with a simple Markovian *environment* of which the state determines the number of newly generated messages in a slot. Specifically, the user population can be in one of two possible environment states, say 1 and 0. During the 1-slots, the number of new messages is given by a random variable which we assume to have a value of no less than 1. As such, the 1-periods represent bursts of high user activity, in which a lot of new messages (at least one per slot) are initiated. During the 0-slots on the other hand, the activity of the users is rather low and typically, the generation of one or more messages per slot happens only once in a while. That is, during each 0-slot the number of newly initiated messages per slot is given by a random variable which can be 0 with finite probability. Translated to the packet stream, this means that in the 1-periods, at least one packet arrives to the multiplexer per slot and the queue content cannot decrease, i.e. the queue is in an overload situation. Of course, if the system is stable, this must be compensated for during the 0-periods, in which slots occur without packet arrivals. As we will show in this chapter, the (*secondary*) correlation introduced in the packet stream by this environment mechanism has a major impact on the multiplexer performance, at least for what the queue content is concerned.

For the model explained above, we also want to investigate the delay and the transmission time experienced by an arbitrary message during equilibrium. To this end, we make the following assumptions regarding our model of the single-server queue.

- Firstly, the queueing discipline is '*FIFO for packets*'. This means that the server is completely ignorant of the fact that each packet belongs to a certain message and just transmits the packets in the order they were stored in the queue.
- Secondly, we note that for the analysis of the queue content in terms of packets, the storage order (and hence, the order of transmission) of packets arriving in the same slot was not relevant. However, for the delay analysis, it is crucial to specify the policy used by the multiplexer to decide in which order these simultaneously arriving packets are stored in the queue. In our model, we assume that this happens in *random* order.
- Finally, in the following sections, we shall always define the delay of either packets or messages to consist of an integer number of slots, i.e. we do not consider a packet to be present in the system before the end of the slot in which it arrived. As such, it suffices to observe the system at slot boundaries only.

2.2 Mathematical model and system equations

Since the environment state process is Markovian, the two states 1 and 0 both have geometric sojourn times:

$$\begin{aligned} \text{Prob[length of a 1-period is } n\,] &= (1-\alpha)\,\alpha^{n-1} \quad , n \geqslant 1\,; \\ \text{Prob[length of a 0-period is } n\,] &= (1-\beta)\,\beta^{n-1} \quad , n \geqslant 1. \end{aligned}$$

Moreover, because of the memoryless property of the geometric distribution, knowledge of the value of r_{k-1}, the environment state in slot $k-1$, is sufficient to determine the probability distribution of r_k. Specifically, we have that

$$\begin{aligned} \mathrm{E}[z^{r_k}\,|\,r_{k-1}=1] &= 1-\alpha+\alpha\,z \triangleq r_1(z)\,; \\ \mathrm{E}[z^{r_k}\,|\,r_{k-1}=0] &= \beta+(1-\beta)\,z \triangleq r_0(z)\,, \end{aligned} \tag{32}$$

where $\mathrm{E}[\cdot]$ denotes the expected value operator. For ease of notation, we also define the function $Q(z)$ as

$$Q(z) \triangleq \frac{r_1(z)}{r_0(z)}\,. \tag{33}$$

Instead of by α and β, the user environment can also be characterised by the more comprehensible set of parameters σ and K, which are defined as follows. Suppose the environment state is 1 with probability σ and 0 with probability $1-\sigma$, independently from slot to slot. The mean sojourn times are then given by $1/(1-\sigma)$ and $1/\sigma$ respectively. It is clear that the overall fraction of 1-slots remains equal to σ if the mean lengths of the 1-periods and the 0-periods are both multiplied by the same factor K, even if the environment is no longer independent then. So, in general, each (α, β)

corresponds to a certain (σ, K) and vice versa, such that

$$\begin{aligned} \mathrm{E}[\,\text{length of 1-period}\,] &= \frac{1}{1-\alpha} = \frac{K}{1-\sigma}\,, \\ \mathrm{E}[\,\text{length of 0-period}\,] &= \frac{1}{1-\beta} = \frac{K}{\sigma}\,. \end{aligned} \tag{34}$$

The factor K can be seen as a measure for the absolute lengths of the sojourn times in both states, whereas the parameter σ characterizes their relative lengths. Therefore, we shall henceforward call K the *burst-length factor*, or equivalently, the *correlation factor* of the environment. Moreover, it turns out that the *correlation coefficient* ϕ between the environment state in two consecutive slots during equilibrium is determined only by the value of K:

$$\phi \triangleq \lim_{k\to\infty} \frac{\mathrm{E}[r_k r_{k+1}] - \mathrm{E}[r_k]\,\mathrm{E}[r_{k+1}]}{\sqrt{\mathrm{Var}[r_k]\,\mathrm{Var}[r_{k+1}]}} = -1+\alpha+\beta = 1 - \frac{1}{K}\,. \tag{35}$$

The message lengths (the number of composing packets) are iid random variables with probability mass function ℓ_j ($j \geqslant 1$) and probability generating function (pgf)

$$L(z) \triangleq \sum_{j=1}^{\infty} \ell_j\, z^j\,.$$

Since the messages arrive to the queue in trains, a new message, or a user becoming active, is seen by the multiplexer as the arrival of a *leading packet* in the current slot and one packet arrival in each of the following consecutive slots up to a total given by the length of the message. We do not impose an upper bound on the message length although it is necessary to assume that the moments of its distribution exist, so any heavy-tailed distributions are precluded. Note that the analysis in this chapter is a priori valid for finite-length messages, i.e. if $L(z) \triangleq \sum_{j=1}^{N} \ell_j\, z^j$ for a certain $N \geqslant 1$, and can be much simplified in this case (see Appendix 2.C). In our further analysis, instead of the probabilities ℓ_n ($n \geqslant 1$), we shall often use the probabilities $q(n)$ ($n \geqslant 1$), which are defined as follows: $q(n)$ denotes the probability that a message that is already n packets long in the current slot, will still continue in the next slot, i.e.,

$$q(n) \triangleq \frac{\sum_{j=n+1}^{+\infty} \ell_j}{\sum_{j=n}^{+\infty} \ell_j} \quad , n \geqslant 1\,. \tag{36}$$

As such, when a user becomes active it runs through a state diagram as the one depicted in Fig. 2.2, which illustrates the meaning of the *incremental* probabilities $q(n), n \geqslant 1$.

Let us now denote by $m_{n,k}$ the number of users that send the nth packet of a message during slot k. The number of leading packet arrivals or newly generated messages $m_{1,k}$ in slot k is assumed to have a probability distribution that only depends on the environment state r_k in slot k. So the variable $m_{1,k}$ depends on the numbers of new messages generated during previous slots *only* through the environment state process.

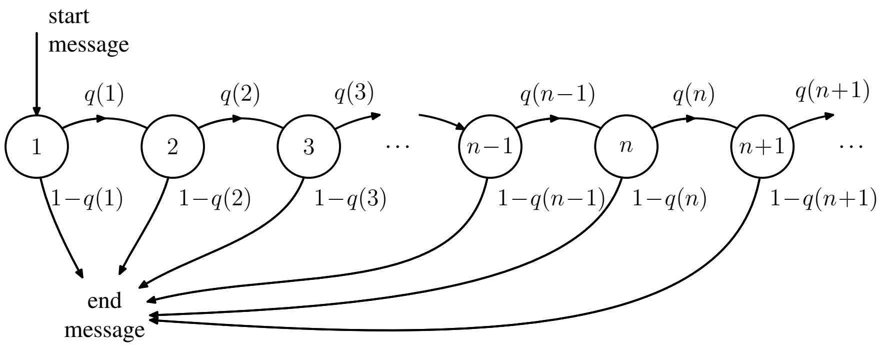


Figure 2.2: State transition diagram of an active user

As such, $m_{1,k}$ is either given by a random variable $w_k^{(1)}$ or by $w_k^{(0)}$, depending on whether the environment is in state 1 or 0, i.e.

$$m_{1,k} = \begin{cases} w_k^{(1)} & \text{if } r_k = 1\,; \\ w_k^{(0)} & \text{if } r_k = 0\,. \end{cases} \tag{37}$$

Both sequences $w_k^{(1)}$ and $w_k^{(0)}$ are iid with pgfs $M_1(z)$ and $M_0(z)$ respectively, again assuming that all needed moments of these distributions are finite,

$$\begin{aligned} M_0(z) &\triangleq \mathrm{E}[\, z^{m_{1,k}} \,|\, r_k = 0\,]\,; \\ M_1(z) &\triangleq \mathrm{E}[\, z^{m_{1,k}} \,|\, r_k = 1\,]\,. \end{aligned} \tag{38}$$

We repeat here the above-mentioned assumption that $M_1(0) = 0$, thus interpreting 1 as the 'active' environment state in which the users generate at least one new message per slot. Again, for ease of notation, we introduce the function $\mu(z)$, defined as

$$\mu(z) \triangleq \frac{M_1(z)}{M_0(z)}\,. \tag{39}$$

Now, for $n \geqslant 1$, the random variable $m_{n+1,k}$ can be expressed as

$$m_{n+1,k} = \sum_{i=1}^{m_{n,k-1}} d_{n,i} \quad , n \geqslant 1\,. \tag{40}$$

The $d_{n,i}$'s ($n \geqslant 1$) in (40) are, for given n, iid random variables with common pgf

$$D_n(z) = q(n)z + 1 - q(n)\,, \qquad n \geqslant 1\,. \tag{41}$$

Indeed, the number of messages that generate their $(n+1)$th packet in slot k corresponds to the number of messages that generate their nth packet during slot $k-1$ and continue in slot k by sending an $(n+1)$th packet. The *total* number of packet arrivals

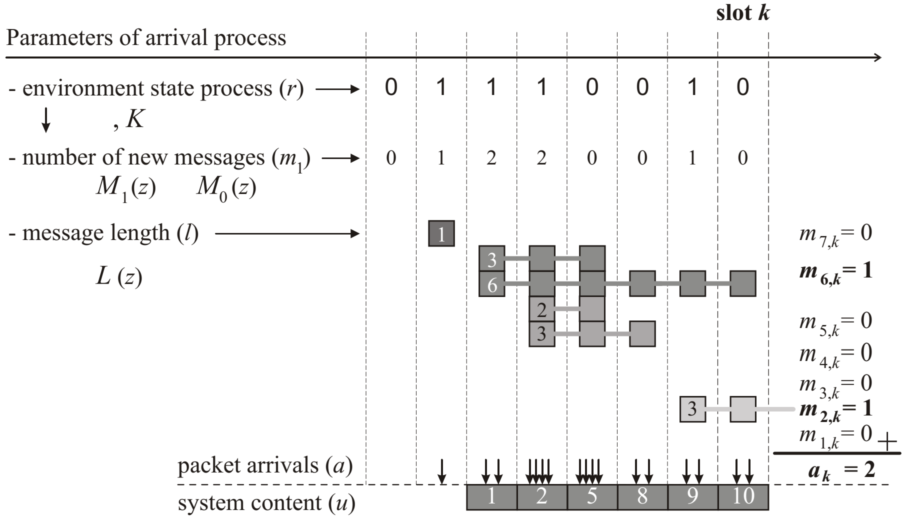


Figure 2.3: Parameters of the arrival process and an example depicting the meaning of the variables $m_{n,k}$.

a_k in slot k is given by the sum of the numbers of active users $m_{n,k}$ ($n \geqslant 1$) present in each of the stages in Fig. 2.2 :

$$a_k = \sum_{n=1}^{+\infty} m_{n,k} \ . \tag{42}$$

Finally, let u_{k+1} represent the *queue content* or *system content* just after slot k (i.e., at the beginning of slot $k+1$), where the terms queue or system content indicate the number of packets that are either waiting or being transmitted. Since the server transmits a packet whenever one is available in the queue, the evolution of the system content from slot to slot is described by

$$u_{k+1} = a_k + (u_k - 1)^+ \,, \tag{43}$$

where $(\cdot)^+ = \max(\cdot, 0)$.

It is now clear that the arrival process of our model is determined by the following parameters, listed in Fig. 2.3: σ and K for the environment, the pgf $L(z)$ of the message length and the pgfs $M_1(z)$ and $M_0(z)$ of the numbers of new messages in a 1-slot and 0-slot respectively. This figure also shows an example explaining the meaning of the random variables $m_{n,k}$ and (42). Fig. 2.4 gives an overview of the packet arrival process as well.

Incremental message length distribution: properties

The pgfs $D_n(z)$ have a number of interesting properties, which will make it possible to revert our analysis later on in this chapter back in terms of the pgf $L(z)$ of the

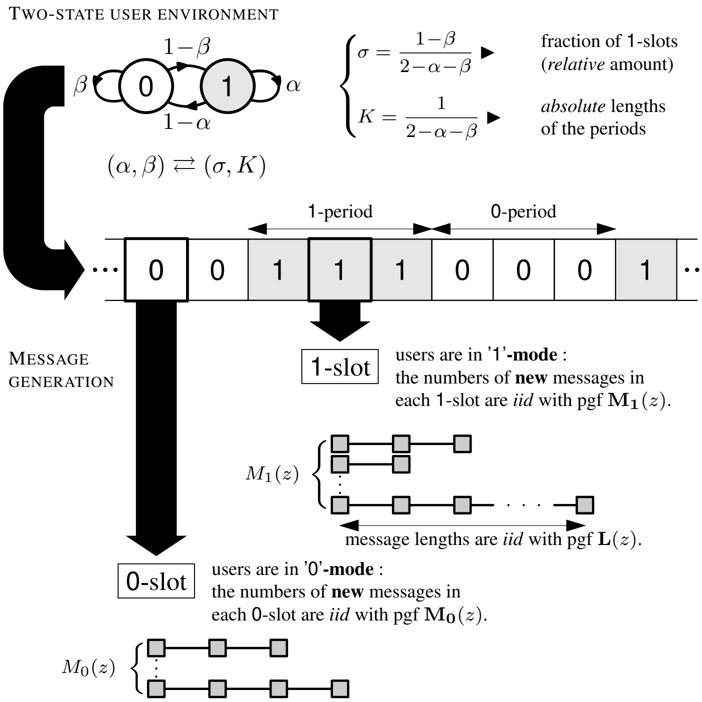


Figure 2.4: Stochastic model of the packet arrival process.

message lengths. Specifically, from the above expression for $D_n(z)$ and equation (36) for $q(n)$, one can derive that for any x and for $n, i \geqslant 1$,

$$D_n(zD_{n+1}(\ldots zD_{n+i-1}(zx)\ldots)) = \frac{\sum_{j=n}^{n+i-1} \ell_j z^{j-n} + \left(1 - \sum_{j=1}^{n+i-1} \ell_j\right) xz^i}{1 - \sum_{j=1}^{n-1} \ell_j} . \tag{44}$$

Hence, for all z for which $L(z)$ converges,

$$\lim_{i\to\infty} D_n(zD_{n+1}(\ldots zD_{n+i-1}(z\,x)\ldots)) = z^{-n} \frac{L(z) - \sum_{j=1}^{n-1} \ell_j z^j}{1 - \sum_{j=1}^{n-1} \ell_j} , \tag{45}$$

and in view of the normalisation equation $L(1) = \sum_{j=1}^{+\infty} \ell_j = 1$,

$$\lim_{i\to\infty} D_n(D_{n+1}(\ldots D_{n+i-1}(x)\ldots)) = 1 .$$

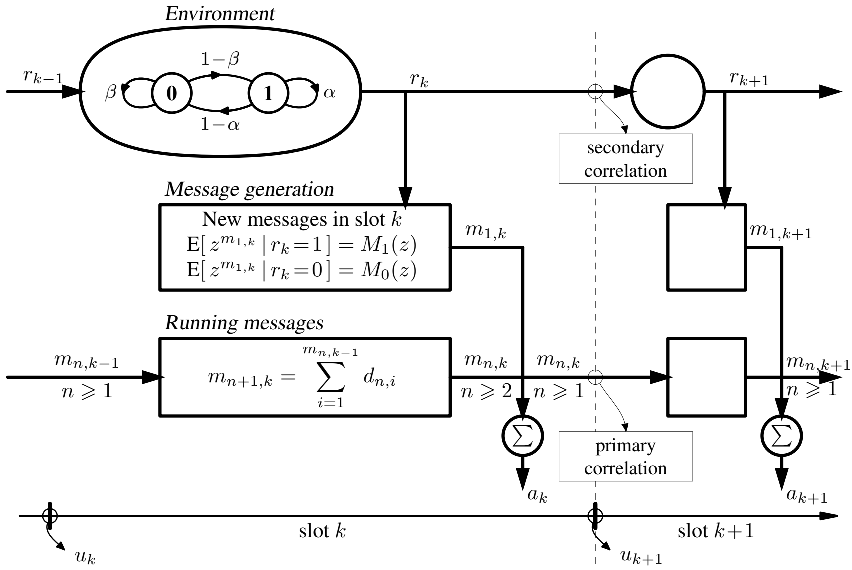


Figure 2.5: Flow chart of the packet arrival process: primary and secondary correlation

2.3 Equilibrium distribution of the system state

In Fig. 2.5, we have used the system equations (32), (37), (40) and (42) to depict the correlation structure of the packet arrival process in the form of a flow chart. First of all, the chart demonstrates the intra-slot dependencies, i.e. how the number of packet arrivals a_k in slot k depends on both the environment state r_k and the numbers of users $m_{n,k}\,(n \geqslant 1)$ that are active in that slot. Moreover, the chart also shows how these quantities determine the corresponding quantities in the *next* slot. Specifically, we see that the slot boundary is crossed by two arrows, corresponding to the two types of correlation mentioned earlier. The first arrow passes the environment state r_k to the next slot and accounts for what we have called the secondary correlation. It is clear from the chart that the environment state process r_k forms a one-dimensional Markov chain in its own right. The second arrow passes the value of the variables $m_{n,k}\,(n \geqslant 1)$ to the next slot and accounts for the primary correlation. Indeed, because the distribution of the message length is general, we have to keep track of the numbers of users that became active in each of the previous slots and still have a packet to send during the current slot. The information passed by both arrows together gives a full description of the state of the packet arrival process in slot k. Therefore, the vectors $\{r_k,\, m_{n,k}(n \geqslant 1)\}$ form an (infinite-dimensional) Markov chain by which the packet arrival process is characterized. In fact, if we consider the system as a whole (not only the arrival process), we should include a third arrow, which passes the queue content u_{k+1} to the next slot and accounts for the relation (43). All three arrows together, i.e. the vector $\{r_k,\, m_{n,k}(n \geqslant 1),\, u_{k+1}\}$, is sufficient to fully describe the state of the multiplexer after slot k. Therefore, we can conclude that these vectors — again

— form an infinite-dimensional Markov chain, and we shall often refer to the set of variables r_k, $m_{n,k}$ $(n \geqslant 1)$, u_{k+1} as the *system state variables* for slot k. Following the outline of the discrete supplementary variable technique (DSVT) described in Sec. 1.3.4, this concludes the step where a suitable Markovian system state description is identified.

Now, we introduce the joint pgf of the infinite-dimensional system state vector in a slot. Because the state vectors form a Markov chain, it is possible to express the pgf of the system state vector in a certain slot as a function of the pgf corresponding to the previous slot, by using the system equations derived earlier. This relation then results in a *functional equation* for the equilibrium joint pgf of the system state vector. Although we are not able to solve this equation, we can derive various results from it regarding the equilibrium marginal distributions, such as the moments and tail distribution of the queue content and the packet delay, which is done in Sec. 2.4. Later, in Sec. 2.5 we use the functional equation to obtain the mean of the delay and transmission time experienced by an arbitrary message as well as the tail distribution of the message delay.

We start the analysis by defining the joint pgf of the system state vector in slot k (i.e. of the random variables r_k, $m_{n,k}(n \geqslant 1)$ and u_{k+1}) as

$$P_k(x,\, y_1,\, y_2,\, \ldots\,,\, z) \triangleq \mathrm{E}\left[\, x^{r_k} \cdot y_1^{m_{1,k}} \cdot y_2^{m_{2,k}} \cdot \ldots \cdot z^{u_{k+1}} \,\right] . \tag{46}$$

Then, using the system equations introduced in the previous section, we find

$$\begin{aligned} P_{k+1}(x, y_1, \ldots, z) \;&=\; M_0(y_1 z)\, r_0\!\left(x\, \frac{M_1(y_1 z)}{M_0(y_1 z)}\right) \\ &\cdot \mathrm{E}\!\left[\left(\frac{r_1\left(x\,\frac{M_1(y_1 z)}{M_0(y_1 z)}\right)}{r_0\left(x\,\frac{M_1(y_1 z)}{M_0(y_1 z)}\right)}\right)^{r_k}\Big(D_1(y_2 z)\Big)^{m_{1,k}}\Big(D_2(y_3 z)\Big)^{m_{2,k}} \cdot \ldots \cdot z^{(u_{k+1}-1)^+}\right]. \end{aligned} \tag{47}$$

In order to remove the operator $(\cdot)^+$ and to express the right-hand side of (47) in terms of the function P_k, we need to distinguish between two cases, namely $u_{k+1} > 0$ and $u_{k+1} = 0$. Doing so, while expanding the E-operator, we get

$$\begin{aligned} P_{k+1}(x, y_1, \ldots, z) = M_0(y_1 z)\, r_0\!\left(x\, \frac{M_1(y_1 z)}{M_0(y_1 z)}\right) & \left[\sum_{i,j_n} \sum_{h=1}^{+\infty} \left(\frac{r_1\left(x\,\frac{M_1(y_1 z)}{M_0(y_1 z)}\right)}{r_0\left(x\,\frac{M_1(y_1 z)}{M_0(y_1 z)}\right)}\right)^{i}\right. \\ &\cdot \prod_{n=1}^{+\infty} \Big(D_n(y_{n+1} z)\Big)^{j_n} \cdot z^{h-1} \cdot \mathrm{Prob}[r_k = i, m_{n,k} = j_n (n \geqslant 1), u_{k+1} = h] \\ &+ \sum_{i,j_n} \left(\frac{r_1\left(x\,\frac{M_1(y_1 z)}{M_0(y_1 z)}\right)}{r_0\left(x\,\frac{M_1(y_1 z)}{M_0(y_1 z)}\right)}\right)^{i} \\ &\left.\cdot \prod_{n=1}^{+\infty} \Big(D_n(y_{n+1} z)\Big)^{j_n} \cdot \mathrm{Prob}[r_k = i, m_{n,k} = j_n (n \geqslant 1), u_{k+1} = 0]\right] \end{aligned}$$

$$= M_0(y_1z)\, r_0\Big(x\,\frac{M_1(y_1z)}{M_0(y_1z)}\Big)\left[\frac{1}{z}P_k\left(\frac{r_1\left(x\,\frac{M_1(y_1z)}{M_0(y_1z)}\right)}{r_0\left(x\,\frac{M_1(y_1z)}{M_0(y_1z)}\right)}, D_1(y_2z), D_2(y_3z), \ldots, z\right)\right.$$
$$+\,\frac{z-1}{z}\sum_{i,j_n}\left(\frac{r_1\left(x\,\frac{M_1(y_1z)}{M_0(y_1z)}\right)}{r_0\left(x\,\frac{M_1(y_1z)}{M_0(y_1z)}\right)}\right)^i$$
$$\left.\cdot\prod_{n=1}^{+\infty}\big(D_n(y_{n+1}z)\big)^{j_n}\cdot \mathrm{Prob}[r_k=i, m_{n,k}=j_n(n\geqslant 1), u_{k+1}=0]\right]. \tag{48}$$

Finally, we note that having an empty system at the beginning of slot $k+1$ implies that no packets have entered the system in slot k. Since we assumed that at least one leading packet arrives in the system during a 1-slot, the system can only be empty at the start of slot $k+1$ when $r_k=0$ and $a_k=0$. Hence, in view of (42), if $u_{k+1}=0$, then also $r_k=0$ and $m_{n,k}=0$ $(n\geqslant 1)$, which implies that the probabilities on the right of (48) are zero if i and j_n $(n\geqslant 1)$ are not all equal to zero. Furthermore, as k goes to infinity, the functions P_k and P_{k+1} converge to the same limiting function P and it is seen from (48) that the equilibrium joint pgf $P(x, y_1, y_2, \ldots, z)$ must satisfy the following functional equation:

$$\begin{aligned} P(x, y_1, y_2, \ldots, z) =\;& \frac{M_0(y_1z)\, r_0\big(x\mu(y_1z)\big)}{z} \\ &\cdot\big[P\left(Q\big(x\,\mu(y_1z)\big), D_1(y_2z), D_2(y_3z), \ldots, z\right) + p_0\,(z-1)\big], \end{aligned} \tag{49}$$

where we have used the shorthand notations (33) and (39). The quantity p_0 indicates the equilibrium probability of having an empty queue at the start of an arbitrary slot. This probability can be found using the normalisation condition $P(1,1,\ldots,1)=1$, although this does not work if (49) is used directly. Let us defer the explanation of how to obtain p_0 to Sec. 2.4.2, where we find

$$p_0 \;=\; \mathrm{Prob}[u=0] \;=\; 1-\big(\sigma M_1'(1) + (1-\sigma)M_0'(1)\big)L'(1)\,. \tag{50}$$

Generally, finding an explicit solution for the joint equilibrium pgf $P(x, y_1, y_2, \ldots, z)$ from the functional equation (49) is not a straightforward task. Nevertheless, in Appendix 2.A we show how a recursive solution can be obtained. But even without an explicit solution for the joint pgf P, the moments of the performance measures we are interested in can be obtained directly from the functional equation (49), as will be explained in Secs. 2.4 and 2.5.

2.4 Analysis in terms of packets

2.4.1 The stability condition

First of all, equation (49) lets us obtain the equilibrium pgf $R(x)$ corresponding to the environment state process if we let all arguments of the P-function other than x be equal to one:

$$R(x) \triangleq P(x,1,1,\ldots,1) \;=\; r_0(x)\,R\!\left(\frac{r_1(x)}{r_0(x)}\right) \;=\; r_0(x)\,R\big(Q(x)\big)\,. \tag{51}$$

Keeping in mind that $R(x)$ is a polynomial of degree one, we get

$$R(x) \;=\; \frac{(1-\beta)\,x+1-\alpha}{2-\alpha-\beta} \;=\; \sigma\,x+1-\sigma\,. \tag{52}$$

In the previous section, we have assumed that the system reaches stochastic equilibrium for $k\to\infty$. However, this is not true for all possible configurations of the packet arrival process. An obvious restriction on the parameters in order for our present analysis to be valid is this: on average over a long period of time, no more packets may enter the system than can possibly be served in that time. As we assume our models in this thesis to be ergodic, this condition simply states that the expected number of arrivals $\mathrm{E}[a]$ per slot must be smaller than 1.

Now, to calculate $\mathrm{E}[a]$, we can proceed as follows. Let $\mathcal{A}_n(y_n)$ ($n\geqslant 1$) be the pgf of the number of users m_n sending the nth packet of a message during an arbitrary slot during equilibrium. For $n=1$, it is clear from (37) and (52) that the pgf $\mathcal{A}_1(y_1)$ of the number of *leading packet* arrivals m_1 is given by

$$\begin{aligned}\mathcal{A}_1(y_1) \;&=\; M_1(y_1)\,\mathrm{Prob}[r=1] \;+\; M_0(y_1)\,\mathrm{Prob}[r=0]\\ &=\; \sigma\,M_1(y_1)+(1-\sigma)M_0(y_1)\,.\end{aligned} \tag{53}$$

For $n\geqslant 2$, on the other hand, we obtain from (40)

$$\mathcal{A}_n(y_n) \;=\; \mathrm{E}[y_n^{m_n}] \;=\; \mathrm{E}\big[\big(D_{n-1}(y_n)\big)^{m_{n-1}}\big] \;=\; \mathcal{A}_{n-1}\big(D_{n-1}(y_n)\big)\,. \tag{54}$$

Upon repeatedly applying this recurrence equation, it is seen that

$$\begin{aligned}\mathcal{A}_n(y_n) \;&=\; \mathcal{A}_1\big(D_1(D_2(D_3(\ldots D_{n-1}(y_n)\ldots)))\big)\\ &=\; \mathcal{A}_1\Big(y_n+(1-y_n)\sum_{j=1}^{n-1}\ell_j\Big)\,,\end{aligned} \tag{55}$$

where we have used property (44) with $z=1$, $n=1$ and $x=y_n$. The moment-generating property of pgfs states that the mean value of a random variable is given by the first derivative of its pgf of which the argument is put equal to 1. Hence, the average number of users that deliver the nth packet of a message during equilibrium is given by

$$\mathrm{E}[m_n] \;=\; \mathcal{A}_n'(1) \;=\; \mathcal{A}_1'(1)\Big(1-\sum_{j=1}^{n-1}\ell_j\Big)\,,\; n\geqslant 1\,. \tag{56}$$

Now, the expected value $\mathrm{E}[a]$ of the total number of packet arrivals can be found by summing all $\mathrm{E}[m_n]$ ($n\geqslant 1$). From (56) and (53), we find that

$$\mathrm{E}[a]=\sum_{n=1}^{+\infty}\mathcal{A}_n'(1)=\mathcal{A}_1'(1)\,L'(1)=\big(\sigma\,M_1'(1)+(1-\sigma)M_0'(1)\big)L'(1)\,. \tag{57}$$

Because each packet requires exactly one slot of service and the multiplexer has only one server, the quantity $\mathrm{E}[a]$ also equals the load ρ of the system, and the equilibrium condition is given by

$$\rho \,=\, \mathrm{E}[a] \,=\, \mathcal{A}_1'(1)L'(1) \,<\, 1\,. \tag{58}$$

2.4.2 Moments of the queue content

Let us now concentrate on the system content u at the beginning of an arbitrary slot during equilibrium. In this section, we show that it is possible to calculate $\mathrm{E}[u]$, $\mathrm{Var}[u]$ and even higher-order moments of u directly from the functional equation (49), i.e. without first requiring an explicit expression for the distribution of u.

To obtain $\mathrm{E}[u]$, we can proceed as follows. First, let us consider (49) for only those values of x, y_n $(n \geqslant 1)$ and z for which the respective arguments of the functions P on both sides of the equation are equal to each other, i.e., the values given by the equations

$$y_n \,=\, D_n(y_{n+1}z) \quad ,n \geqslant 1\,; \tag{59}$$

$$x \,=\, \frac{r_1(x\,\mu(y_1 z))}{r_0(x\,\mu(y_1 z))} \,=\, Q\big(x\,\mu(y_1 z)\big)\,. \tag{60}$$

The above equations appear to have more than one set of solutions. For our purposes, it is sufficient to consider only the solutions $x \triangleq \chi(z)$ and $y_n \triangleq \eta_n(z)$, which satisfy $\chi(1) = 1$ and $\eta_n(1) = 1$ $(n \geqslant 1)$. By repeatedly applying (59) and using property (45), we then find for $n \geqslant 1$

$$\eta_n(z) \,=\, \lim_{N\to\infty} D_n(zD_{n+1}(\ldots zD_N(z)\ldots)) \,=\, \frac{L(z) - \sum\limits_{j=1}^{n-1} \ell_j\, z^j}{z^n\left(1 - \sum\limits_{j=1}^{n-1} \ell_j\right)}. \tag{61}$$

Note in particular that $\eta_1(z) = L(z)/z$. From this result, together with equation (60), it follows that the function $\chi(z)$ is determined implicitly by

$$\chi(z) \,=\, \frac{r_1(\chi(z)\,\mu(L(z)))}{r_0(\chi(z)\,\mu(L(z)))}\;; \qquad \chi(1) = 1\,. \tag{62}$$

When the arguments $(x, y_1, y_2, \ldots, z)$ are confined to the one-dimensional subspace ($\chi(z)$, $\eta_1(z)$, $\eta_2(z)$,...,z), equation (49) becomes a linear equation for the function $P(\chi(z), \eta_1(z), \eta_2(z), \ldots, z)$, from which we get

$$P(\chi(z), \eta_1(z), \eta_2(z), \ldots, z) \,=\, \frac{p_0\,(z-1)\,G(z)}{z - G(z)}\,, \tag{63}$$

with

$$G(z) \,\triangleq\, M_0\big(L(z)\big)\, r_0(\chi(z)\mu(L(z)))\,. \tag{64}$$

The quantity p_0 can now be calculated from the normalisation condition

$$P(\chi(z), \eta_1(z), \ldots, z)\big|_{z=1} \,=\, 1\,,$$

as

$$p_0 = 1-\rho\,,$$

where ρ is the load of the multiplexer, given by (58). Note that this agrees with (50). Next, total differentiation of both sides of (63) with respect to z and evaluation of the result at $z=1$ yields

$$1-p_0 + \frac{G''(1)}{2p_0} = \underbrace{\frac{\partial}{\partial x}P(1,\ldots,1)}_{R'(1)}\cdot\chi'(1) + \sum_{n=1}^{+\infty}\Big(\underbrace{\frac{\partial}{\partial y_n}P(1,\ldots,1)}_{\mathcal{A}'_n(1)}\cdot\eta'_n(1)\Big) + \underbrace{\frac{\partial}{\partial z}P(1,\ldots,1)}_{\mathrm{E}[u]}\,.$$

Except for $\mathrm{E}[u]$, all the quantities in this equation can easily be calculated from previous results; together with (52), (53), (56), (61), (62) and (64), we finally obtain:

$$\begin{aligned}\mathrm{E}[u] &= \mathcal{A}'_1(1)L'(1) + \frac{\big(\mathcal{A}'_1(1)\big)^2 L''(1) + \mathcal{A}''_1(1)L'(1)}{2(1-\rho)}\,L'(1) \\ &\quad + \sigma\,(K-1)\big(M'_1(1)-M'_0(1)\big)L'(1)\left[(1-\sigma)\frac{\big(M'_1(1)-M'_0(1)\big)L'(1)}{1-\rho} - 1\right].\end{aligned} \tag{65}$$

It has been verified that this result for $\mathrm{E}[u]$ reduces to the known result [207], if abstraction is made of the environment state process by putting $\alpha=0$ and $\beta=1$, i.e., if the number of new messages is determined by the same pgf $M_0(z)$ in every slot.

Recall that the packet arrival process, and hence the behaviour of the multiplexer, is completely specified when a particular choice for each of the following is made: the parameters σ and K for the environment, the distribution $L(z)$ of the message length ℓ and the distributions $M_1(z)$ and $M_0(z)$ of the numbers of new messages $w^{(1)}$ and $w^{(0)}$ respectively. In order to show more clearly how the mean system content is influenced by these choices, we rewrite (65) as

$$\begin{aligned}\mathrm{E}[u] &= \frac{\sigma(M'_1(1)-M'_0(1))(M'_1(1)L'(1)-1)L'(1)}{1-\rho}\cdot K + \frac{\mathcal{A}'_1(1)^2 L'(1)}{2(1-\rho)}\cdot \mathrm{Var}[\ell] \\ &\quad + \frac{\sigma L'(1)^2}{2(1-\rho)}\cdot\mathrm{Var}[w^{(1)}] + \frac{(1-\sigma)L'(1)^2}{2(1-\rho)}\cdot\mathrm{Var}[w^{(0)}] \\ &\quad + \frac{L'(1)}{2}\bigg[\mathcal{A}'_1(1)\big(2-L'(1)\big) \\ &\qquad + \sigma\big(M'_1(1)-M'_0(1)\big)\bigg(2 - \frac{(1-\sigma)\big(M'_1(1)-M'_0(1)\big)L'(1)}{1-\rho}\bigg)\bigg],\end{aligned} \tag{66}$$

where $\mathcal{A}'_1(1) = \sigma M'_1(1) + (1-\sigma)M'_0(1)$.

First of all, we see that for given distributions of ℓ, $w^{(1)}$ and $w^{(0)}$, and for a given fraction σ of 1-slots, the mean queue content increases linearly with the correlation factor K of the environment state process, this in spite of the fact that the mean number of packet arrivals in a slot is independent of K. An intuitive explanation for this effect can readily be found. Since we assumed that at least one packet enters the system per slot while the environment is in state 1, the server cannot keep up with the arrival stream during the 1-periods and the queue content will gradually increase. Of course, in stochastic equilibrium, these periods of buffer *accumulation* must alternate with periods during which the *transmission* of packets prevails over the arrival of packets and the queue content diminishes again. On the average, such is the case for the 0-periods. As explained above, the mean lengths of both the 1-periods (accumulation) and the 0-periods (transmission) increase linearly with the factor K. Now, for a given value of σ, one sees that the queue content, and also its expected value, will reach higher values as the accumulation periods last longer and are less interrupted by transmission periods, i.e., as K becomes larger. This explains the observed impact of K on $\mathrm{E}[u]$.

Next, we notice that $\mathrm{E}[u]$ is a linearly increasing function of the variances of both the message length and the numbers of new messages per slot in each of the environment states. Therefore, the mean queue content increases as the packet arrival process exhibits a greater 'variability', which is a common fact in queueing theory. Also, for a given load $\rho = \mathcal{A}_1'(1)L'(1)$, the influence of Var$[\ell]$ on $\mathrm{E}[u]$ increases as $L'(1)$ diminishes, whereas the influence of Var$[w^{(1)}]$ and Var$[w^{(0)}]$ decreases. In other words, for a given ρ, a larger number of, hence, shorter messages results in a larger impact of Var$[\ell]$ and a smaller impact of Var$[w^{(1)}]$ and Var$[w^{(0)}]$ on the mean queue content.

In the same way as for $\mathrm{E}[u]$, closed-form results can be derived for the higher-order moments of u in a recursive way. However, it should be noted that the mathematical manipulations rapidly become very complicated. To obtain the nth order moment of u, we differentiate (63) n times and evaluate the result in $z=1$. This yields an expression for the nth partial derivative (PD) of $P(x, y_1, y_2, \ldots, z)$ with respect to z in the point $(1, 1, \ldots, 1)$ as a function of all other (mixed) PDs of $P(x, y_1, y_2, \ldots, z)$ up to order n in that point. The first PD equals the nth derivative of the pgf $U(z)$ of u in $z=1$ and is directly related to the moment $\mathrm{E}[u^n]$ we are looking for. The other (mixed) PDs of $P(x, y_1, y_2, \ldots, z)$ up to order $n-1$ in the point $(1, 1, \ldots, 1)$ were already obtained during the calculation of a previous moment of u. To calculate the remaining nth order PDs of $P(x, y_1, y_2, \ldots, z)$, we can use the functional equation (49) by differentiating both sides n times to the desired arguments and evaluating for $(1, 1, \ldots, 1)$. To show that this is always possible, we can remark the following. When comparing in (49) the arguments of P on the right-hand side to the ones on the left, we observe that the occurrences of y_n $(n \geqslant 1)$ have all shifted one place towards the left. It is this property that enables us to overcome the infinite dimensionality of P and to calculate its consecutive PDs from (49) in a recursive way. Following the above scheme, we have obtained an explicit expression for Var$[u]$. We shall use this expression in the numerical examples of the next section; however, it is too elaborate to be printed here in detail.

We shall need some of the PDs mentioned here later on for the calculation of the mean message delay and transmission time in Secs. 2.5.1 and 2.5.2. In particular, we need these PDs of $(\partial/\partial y_1)P$:

$$\frac{\partial^2}{\partial x \partial y_1} P(1, \ldots, 1) = \sigma M_1'(1)\,; \tag{67}$$

$$\frac{\partial^2}{\partial y_1^2} P(1,\ldots,1) = \mathcal{A}_1''(1)\,;$$

$$\frac{\partial^2}{\partial y_i \partial y_1} P(1,\ldots,1) =$$
$$\Big[\big(\mathcal{A}_1'(1)\big)^2 + \sigma\,(1-\sigma)\big(M_1'(1) - M_0'(1)\big)^2 \phi^{i-1}\Big] \Big(\sum_{j=i}^{+\infty} l_j\Big) \quad , i \geqslant 2\,; \tag{68}$$

$$\frac{\partial^2}{\partial z \partial y_1} P(1,\ldots,1) =$$
$$\mathcal{A}_1''(1) + \mathcal{A}_1'(1)\big[1 - \mathcal{A}_1'(1) + \mathrm{E}[u]\big] + \sigma\,(K-1)(M_1'(1) - M_0'(1)) \tag{69}$$
$$\times \left[M_1'(1) - 1 + \mathcal{A}_1'(1)\big(L'(1) - 1\big) + (1-\sigma)K\big(M_1'(1) - M_0'(1)\big)\big(1 - \frac{L(\phi)}{\phi}\big)\right].$$

2.4.3 Tail behaviour of the queue content

To assess the tail behaviour of the queue content, we must first construct an expression for the pgf $U(z)$ of the queue content u. In appendix 2.A, we give an iterative procedure that yields the joint pgf of the system state that we can use. In view of definition (46), we have that $U(z) = \mathrm{E}[z^u] = P(1,1,\ldots,1,z)$ and it therefore follows from (132) that

$$\begin{aligned} U(z) &= P(1,1,\ldots,1,z) \\ &= p_0(z-1)\sum_{j=0}^{+\infty}\left(\prod_{n=0}^{j}\zeta_n(z)\right) + \frac{p_0(z-1)G(z)}{z-G(z)}\left(\prod_{n=0}^{+\infty}\zeta_n(z)\right), \end{aligned} \tag{70}$$

where

$$\zeta_n(z) = \frac{1}{z}\, M_0\big(\tilde{g}_n(z)\big)\, r_0\big(\tilde{\gamma}_n(z)\big)\,,\; n \geqslant 0\,. \tag{71}$$

In (70) and (71), we denoted

$$\tilde{g}_n(z) \triangleq g_n(1,z) \;=\; zD_1(zD_2(\ldots zD_n(z)\ldots)), \qquad n \geqslant 0\,, \tag{72}$$

which according to (44) is a known polynomial of degree $n+1$ converging to $L(z)$ as n goes to infinity. It is also equal to the function $g_n(y_{n+1}, z)$ defined in (128) but with all other arguments than z taken equal to 1. Likewise, the function $\tilde{\gamma}_n(z)$ is equal to the function γ_n in (129) but also with arguments $x, y_1, \ldots, y_{n+1}$ equal to 1. The functions $\tilde{\gamma}_n(z)$ are therefore defined recursively as:

$$\tilde{\gamma}_n(z) \;\triangleq\; \gamma_n(1,1,\ldots,1,z) \;=\; \begin{cases} \mu(z) & ,n=0\,; \\ Q(\tilde{\gamma}_{n-1}(z))\cdot\mu(\tilde{g}_n(z)) & ,n \geqslant 1\,. \end{cases} \tag{73}$$

To derive the tail distribution of the queue content, we use an approximation technique [43, 46] which is known to yield very accurate results. Specifically, from the inversion formula for z-transforms, it follows that the probability mass function $u(n)$ can be expressed as a weighted sum of negative nth powers of the poles of $U(z)$. Since all these poles have a modulus larger than 1, $u(n)$ is dominated by the contribution of the pole z_0 with the smallest modulus. It is shown [46] that this 'dominant' pole z_0 must necessarily be real and positive in order to ensure a nonnegative probability mass function $u(n)$. As such, the probability of having more than S packets in the system at the beginning of an arbitrary slot can be expressed by the following geometric form for sufficiently large values of S:

$$\mathrm{Prob}[\,u > S\,] \cong -\frac{\theta}{z_0 - 1}\left(\frac{1}{z_0}\right)^{S+1}, \tag{74}$$

where θ is the residue of $U(z)$ in the point $z = z_0$.

To identify z_0 and θ, we can proceed as follows. Since $\lim_{n\to\infty} \tilde{g}_n(z) = L(z)$, it is clear from (73), that if $\tilde{\gamma}_n(z)$ converges to some limiting function $\tilde{\gamma}_\infty(z)$, then this function satisfies

$$Q(\tilde{\gamma}_\infty(z)) \,=\, \frac{r_1(Q(\tilde{\gamma}_\infty(z)) \,\cdot\, \mu(L(z)))}{r_0(Q(\tilde{\gamma}_\infty(z)) \,\cdot\, \mu(L(z)))}\;; \qquad Q(\tilde{\gamma}_\infty(1)) \,=\, 1\,. \tag{75}$$

Comparing (62) and (75), it can thus be concluded that $r_1(\tilde{\gamma}_\infty(z))/r_0(\tilde{\gamma}_\infty(z))$ equals $\chi(z)$, because both functions are implicitly defined by the same relations. Hence, from (75), it also follows that

$$\tilde{\gamma}_\infty(z) \,=\, \chi(z)\cdot\mu(L(z))\,. \tag{76}$$

Although we strongly believe $\tilde{\gamma}_n(z)$ to converge in n for all relevant complex values of z, we can only prove this if z is real and strictly positive. Hence, for this range of z-values (which includes the dominant pole z_0), expression (70) is certain to converge. Also, from the specific structure of (70), we see that any real root $z^\star$ of $z - G(z) = 0$ larger than 1 is a pole of $U(z)$. It is not difficult to show that there is just one such root and that it has multiplicity 1. Moreover, we can prove that any real pole of the factors $\zeta_n(z)$ ($n \geqslant 0$) must exceed $z^\star$. Hence, the root $z^\star$ is equal to the real dominant pole z_0 of $U(z)$ and can be calculated numerically by using, for instance, the Newton-Raphson algorithm.

The residue θ can now be determined from (70) as

$$\theta \,=\, \operatorname*{Res}_{z=z_0} U(z) \,=\, \left(\prod_{n=0}^{+\infty} \zeta_n(z_0)\right)\cdot\frac{p_0(z_0-1)z_0}{1-G'(z_0)}\,. \tag{77}$$

Considering definition (64) of $G(z)$ and the limit behaviour of both $\tilde{g}_n(z)$ and $\tilde{\gamma}_n(z)$, we observe that $M_0(\tilde{g}_n(z_0))\, r_0(\tilde{\gamma}_n(z_0))$ goes to $G(z_0) = z_0$ as n approaches infinity. From (71) it then follows that the factors $\zeta_n(z_0)$ of the infinite product in (77) go to 1, as n goes to infinity. As a consequence, (77) allows us to compute the residue θ up to any desired precision by taking into account a sufficient number of factors.

2.4.4 Moments of packet arrivals per slot

We have called a the total number of packet arrivals per slot during equilibrium, which is the sum of all m_n ($n \geqslant 1$) as given by (42). The mean value of a was calculated earlier, when discussing the stability condition of the system, as (57). Now, for use in the following section, we would also like to have an expression for the variance of a. One 'natural' way to obtain the moments of a would be to establish its pgf $A(y)$ first and then use the moment-generating property of this pgf. Note that $A(y)$ cannot be constructed by merely multiplying the pgfs $\mathcal{A}_n(y)$ of m_n ($n \geqslant 1$), since these variables are not independent. Instead, we could observe that $A(y) = P(1, y, \ldots, y, 1)$ and repeatedly apply (49), much like we have done for the pgf of the queue content.

Alternatively, in view of (42), the second-order moment of a can be calculated as

$$\mathrm{E}[a^2] = \mathrm{E}\Big[\Big(\sum_{n=1}^{+\infty} m_n\Big)^2\Big] = \sum_{n=1}^{+\infty} \mathrm{E}[m_n^2] + 2\sum_{i=1}^{+\infty}\sum_{j=i+1}^{+\infty} \mathrm{E}[m_i \cdot m_j]\,, \tag{78}$$

where

$$\mathrm{E}[m_i \cdot m_j] = \frac{\partial^2}{\partial y_i \partial y_j} P(1,1,\ldots,1) \qquad \text{and} \qquad \mathrm{E}[m_n^2] = \mathcal{A}_n''(1) + \mathcal{A}_n'(1)\,. \tag{79}$$

As explained above, the partial derivatives of $P(x, y_1, y_2, \ldots, z)$ evaluated in $(1, 1, \ldots, 1)$ can be calculated directly from the functional equation (49). Doing so, for the variance $\mathrm{Var}[a] = \mathrm{E}[a^2] - \mathrm{E}[a]^2$ of the number of packet arrivals a during an arbitrary slot we obtain :

$$\begin{aligned}\mathrm{Var}[a] = {} & \big(\mathcal{A}_1''(1) - \mathcal{A}_1'(1)^2\big) \sum_{i=1}^{+\infty} \Big(1 - \sum_{h=1}^{i-1} \ell_h\Big)^2 + \mathcal{A}_1'(1)\, L'(1) \\ & + 2\sigma(1-\sigma)\big(M_1'(1) - M_0'(1)\big)^2 \sum_{i=1}^{+\infty}\sum_{j=i+1}^{+\infty} \phi^{j-i}\Big(1 - \sum_{h=1}^{i-1}\ell_h\Big)\Big(1 - \sum_{h=1}^{j-1}\ell_h\Big)\,.\end{aligned} \tag{80}$$

2.4.5 Mean packet delay

We define the packet delay as the number of slots between the end of the packet's arrival slot and the end of the slot during which the packet is transmitted from the queue. Let d with pgf $D(z)$ be the random variable denoting the delay of an arbitrary packet during equilibrium. In [43, 211] and for a more general case also in [193], it is shown that for any discrete-time single-server queueing system, with a FIFO queueing discipline and constant service times of one slot, the following relationship exists between $D(z)$ and the pgf $U(z)$ of the queue content u at the start of an arbitrary slot during equilibrium:

$$D(z) = \frac{U(z) - U(0)}{1 - U(0)}\,, \tag{81}$$

irrespectively of the (possibly correlated) nature of the packet arrival process. Since in our model the above conditions are met, expression (81) for the distribution of the packet delay applies and allows us to obtain the moments and tail distribution of d directly from the corresponding results for u. For instance, the mean packet delay relates to the mean queue content obtained in (65) as ($U(0)=p_0$)

$$\mathrm{E}[\,d\,] = D'(1) = \frac{\mathrm{E}[u]}{1-p_0} = \frac{1}{\big((1-\sigma)M_0'(1) + \sigma\, M_1'(1)\big)\, L'(1)}\mathrm{E}[u]\,, \tag{82}$$

in accordance with Little's theorem [83, 102].

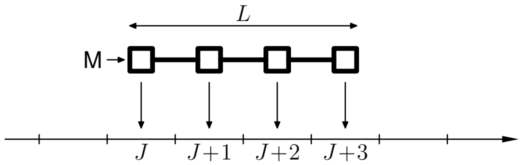


Figure 2.6: An arbitrary message M of length L, with the leading packet arriving in slot J.

2.5 Analysis in terms of messages

2.5.1 Mean message delay

In this section, we deal with the delay experienced by messages, rather than individual packets. We define the *message* delay c as the time period between the end of the slot in which the first packet of a message is generated and the end of the slot during which the last packet of this message is transmitted. In what follows, we extend the technique used in [47,207] to obtain the mean message delay $\mathrm{E}[c]$. To start with, let us first consider an arbitrary but tagged message M, of which the leading packet enters the multiplexer during some slot J during equilibrium, as in Fig. 2.6. Let $r(J+n)$, $m_i(J+n)$ $(i \geqslant 1)$ and $u(J+n)$ be the system state variables for slot $J+n$. Also, by $a(J+n)$, we indicate the total number of packet arrivals during slot $J+n$. Furthermore, if we denote by L the length of message M, then for $0 \leqslant n \leqslant L-1$, one of the packets contained in $a(J+n)$ will be the $(n+1)$th packet of the tagged message M; and by $a^*(J+n)$ we shall count those packets arriving in slot $J+n$ that are to be transmitted no later than this particular packet of M. Now, using similar methods as in [43,45,47], it can be shown that the joint mass function of the system state variables in slot J is given by

$$\begin{aligned}\text{Prob}[r(J)&=i, m_1(J)=j_1, m_2(J)=j_2, \ldots, u(J)=k] \\ &= \frac{j_1}{\mathcal{A}_1'(1)}\text{Prob}[r=i, m_1=j_1, m_2=j_2, \ldots, u=k], \end{aligned} \tag{83}$$

and the corresponding pgf $\widehat{P}(x, y_1, y_2, \ldots, z)$ is given by

$$\widehat{P}(x, y_1, y_2, \ldots, z) = \frac{y_1}{\mathcal{A}_1'(1)} \frac{\partial}{\partial y_1} P(x, y_1, y_2, \ldots, z)\,. \tag{84}$$

From (83) and (84) it is clear that the statistics of the system during equilibrium depend on the events on which one chooses to observe the multiplexer: the joint pgf of the system state 'on slot boundaries' (P) differs from the pgf of the system state 'as seen by new messages' ($\widehat{P}$). In Appendix 2.A, we shall use (84) to give a more explicit expression for $\widehat{P}(x, y_1, y_2, \ldots, z)$.

In order to calculate the mean value of the message delay c, we first condition on the length of M, i.e.

$$\mathrm{E}[c] = \sum_{k=1}^{+\infty} \mathrm{E}[c_k]\,\ell_k\,, \tag{85}$$

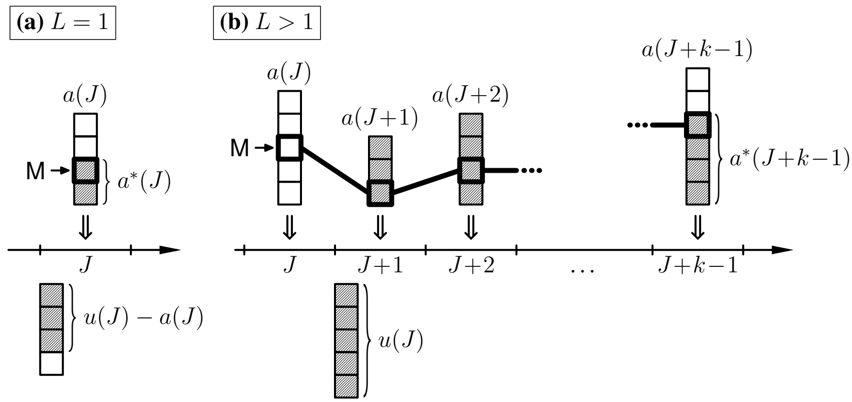


Figure 2.7: Arrival of a message M of length $L = k$. The packets of M are indicated in bold line and the packets that are transmitted during the delay time of M are shaded.

where c_k denotes the delay of an arbitrary message of length k. To derive the conditional mean delay $\mathrm{E}[c_k]$, we make a distinction between the cases for $L = 1$ and for $L > 1$. In Fig. 2.7, we have depicted the packet arrivals in the consecutive slots during which message M is delivered to the queue. Note that we have drawn the arriving packets in the order they are stored in the queue and that in each slot, the position of the packet belonging to M is random (i.e. uniformly distributed). In the case $\underline{L=1}$, it is clear that the message delay is given by

$$c_1 \;=\; u(J) \;-\; a(J) \;+\; a^*(J)\,, \tag{86}$$

which corresponds to the shaded packets in Fig. 2.7**(a)**. If the system is non-empty at the beginning of slot J, then $u(J) - a(J)$ is the number of packets in the system at that moment minus the one packet being served during slot J; otherwise, it is zero. Also, $a^*(J)$ is the number of packet arrivals in slot J to be transmitted no later than the packet of M. Next, if $\underline{L=k>1}$, then the message delay is given by the number of shaded packets in Fig. 2.7**(b)**:

$$c_k \;=\; u(J) \;+\; \sum_{n=1}^{k-2} a(J+n) \;+\; a^*(J+k-1) \quad , \, k > 1\,, \tag{87}$$

where the second term is dropped if $k = 2$, by convention. Due to the fact that the packet belonging to M could be *any* of the $a(J+n)$ packets arriving in slot $J+n$, it can be seen that

$$\begin{aligned} \mathrm{E}[z^{a^*(J+n)}] \;&=\; \frac{z}{z-1} \cdot \mathrm{E}\left[\frac{z^{a(J+n)} - 1}{a(J+n)}\right] \\ \text{and} \quad \mathrm{E}[a^*(J+n)] \;&=\; \frac{1}{2}\,\mathrm{E}[a(J+n)] + \frac{1}{2} \quad , \, n \geqslant 0\,. \end{aligned} \tag{88}$$

Now, the expected values of both (86) and (87) can be substituted into (85), which yields

$$\begin{aligned}\mathrm{E}[c] \;&= \mathrm{E}[u(J)] - \frac{1}{2}\mathrm{E}[a(J)]\,\ell_1 + \frac{1}{2} \\ &\quad + \sum_{k=2}^{+\infty}\sum_{n=1}^{k-1}\mathrm{E}[a(J{+}n)]\,\ell_k - \frac{1}{2}\sum_{k=2}^{+\infty}\mathrm{E}[a(J{+}k{-}1)]\,\ell_k\,. \qquad (89)\end{aligned}$$

The mean queue content $\mathrm{E}[u(J)]$ as seen by an arbitrary message M can be calculated from (84) in terms of the PD of P given by (69):

$$\mathrm{E}[u(J)] \;=\; \frac{\partial}{\partial z}\widehat{P}(1,1,\ldots,1) \;=\; \frac{1}{\mathcal{A}_1'(1)}\,\frac{\partial^2}{\partial z\partial y_1}P(1,1,\ldots,1)\,. \qquad (90)$$

Similarly, to find the mean number of arrivals $\mathrm{E}[a(J)]$ in slot J, we can first use the system equation (42):

$$\mathrm{E}[a(J)] \;=\; \sum_{i=1}^{+\infty}\mathrm{E}[m_i(J)] \;=\; \sum_{i=1}^{+\infty}\frac{\partial}{\partial y_i}\widehat{P}(1,1,\ldots,1)\,.$$

Then, after using (84) again, together with (53) and (68), we find

$$\begin{aligned}\mathrm{E}[a(J)] \;&= \frac{\mathcal{A}_1''(1)+\mathcal{A}_1'(1)-\big(\mathcal{A}_1'(1)\big)^2}{\mathcal{A}_1'(1)} + \mathcal{A}_1'(1)L'(1) \\ &\qquad + \sigma(1{-}\sigma)\,\frac{(M_1'(1)-M_0'(1))^2}{\mathcal{A}_1'(1)}\,\frac{\phi-L(\phi)}{1{-}\phi}\,, \qquad (91)\end{aligned}$$

where ϕ is reminded to be the coefficient of correlation of the environment state process given by (35). Now, the only quantities in (89) that remain to be determined are the mean numbers of arrivals $\mathrm{E}[a(J+n)]$ in the slots $J+n$ ($n \geqslant 1$), given that M is still active in those slots. To this end, we shall first derive in the following paragraphs the expected values of the environment state $r(J+n)$ and of the numbers of users $m_i(J{+}n)$ sending their ith packet ($i{\geqslant}1$). Furthermore, in the remainder of this section, we always assume that $n{\geqslant}1$ and that a packet from message M arrives in slot $J{+}n$.

Environment state in slot $J{+}n$.

From (32), we have for $R_{J+n}(z)$, the pgf of $r(J{+}n)$, that

$$\begin{aligned}R_{J+n}(z) \;&=\; r_1(z)\,\mathrm{Prob}[r(J{+}n{-}1){=}1] + r_0(z)\,\mathrm{Prob}[r(J{+}n{-}1){=}0] \\ &=\; r_0(z)\;R_{J+n-1}\big(Q(z)\big)\,; \qquad (92)\end{aligned}$$

Differentiating both sides of (92) and evaluation in $z{=}1$ yields

$$\mathrm{E}[r(J{+}n)] \;=\; 1{-}\beta \,+\, \mathrm{E}[r(J{+}n{-}1)]\,\phi \;=\; (1{-}\beta)\frac{1-\phi^n}{1{-}\phi} + \mathrm{E}[r(J)]\,\phi^n\,, \qquad (93)$$

where we have recursively applied the first equation to obtain the latter. Next, considering the fact that $\mathrm{E}[r(J)] = (\partial/\partial x)\widehat{P}(1,1,\ldots,1)$, we can conclude from (93), (84) and (67) that

$$\mathrm{E}[r(J{+}n)] \;=\; \sigma \;+\; \sigma(1{-}\sigma)\,\frac{M_1'(1) - M_0'(1)}{\mathcal{A}_1'(1)}\,\phi^n\,. \tag{94}$$

Note that $\mathrm{E}[r(J{+}n)] \neq \sigma$, as would be the case for the environment state in an arbitrary slot. However, slot $J{+}n$ is not just *any* slot, but the nth slot after the arrival of the tagged message M, which is statistically very different! The same remark can be made for the other expressions in this section regarding averages taken in slot $J{+}n$.

New messages in slot $J{+}n$.

Let $\mathcal{A}_{1,J+n}(z)$ be the pgf of $m_1(J{+}n)$. In a similar way as for (92), it follows from (38) that

$$\begin{aligned}\mathcal{A}_{1,J+n}(z) \;&=\; \mathrm{E}[z^{m_1(J+n)}]\\ &=\; \mathrm{E}[M_0(z)\,\big(\mu(z)\big)^{r(J+n)}] \;=\; M_0(z)\,R_{J+n}\big(\mu(z)\big)\,.\end{aligned} \tag{95}$$

Then, after differentiating (95) and using (94), we find

$$\mathrm{E}[m_1(J{+}n)] \;=\; \mathcal{A}_1'(1) + \sigma(1{-}\sigma)\,\frac{(M_1'(1) - M_0'(1))^2}{\mathcal{A}_1'(1)}\,\phi^n\,. \tag{96}$$

Active messages initiated before slot $J{+}n$.

In view of equation (40), it follows that for $i>1$:

$$\begin{aligned} m_i(J{+}n) \;&=\; \sum_{j=1}^{m_{i-1}(J+n-1)} d_{i-1,j} \quad, i \neq n{+}1\,;\\ m_{n+1}(J{+}n) - 1 \;&=\; \sum_{j=1}^{m_n(J+n-1)-1} d_{n,j}\,,\end{aligned} \tag{97}$$

where, for given i, the $d_{i-1,j}$'s are iid with common pgf $D_{i-1}(z)$, given in (41), and with expected value $q(i{-}1)$. The term -1 in (97) accounts for the $(n{+}1)$th packet of M that is certain to arrive in slot $J{+}n$. Taking expected values of (97), we obtain:

$$\begin{aligned}\mathrm{E}[m_i(J{+}n)] \;&=\; q(i{-}1)\,\mathrm{E}[m_{i-1}(J{+}n{-}1)] \quad, 1{<}i \neq n{+}1\,;\\ \mathrm{E}[m_{n+1}(J{+}n)] - 1 \;&=\; q(n)\,\big(\mathrm{E}[m_n(J{+}n{-}1)] - 1\big)\,.\end{aligned}$$

Recursive application of the above equations and using definition (36), then yields

$$\mathrm{E}[m_i(J{+}n)] \;=\; \Big(\sum_{j=i}^{+\infty}\ell_j\Big)\,\mathrm{E}[m_1(J{+}n{-}i{+}1)] \quad, 1 < i < n{+}1\,; \tag{98}$$

$$\mathrm{E}[m_{n+1}(J{+}n)] - 1 \;=\; \Big(\sum_{j=n+1}^{+\infty} \ell_j\Big)\,\big(\mathrm{E}[m_1(J)] - 1\big)\;; \tag{99}$$

$$\mathrm{E}[m_i(J{+}n)] \;=\; \frac{\sum\limits_{j=i}^{+\infty} \ell_j}{\sum\limits_{j=i-n}^{+\infty} \ell_j}\,\mathrm{E}[m_{i-n}(J)] \quad , \, i > n{+}1\,. \tag{100}$$

Now, in (98), the averages on the right-hand side can directly be obtained from (96). On the other hand, in (99) and (100), the quantities $\mathrm{E}[m_i(J)]$ are given by $(\partial/\partial y_i)\widehat{P}(1,$ $1, \ldots, 1)$ for each $i \geqslant 1$ respectively. Then, using (84) and the needed PDs of P, we find

$$\mathrm{E}[m_i(J{+}n)] \;=\; \Big[\mathcal{A}_1'(1) + \sigma(1{-}\sigma)\,\frac{(M_1'(1) - M_0'(1))^2}{\mathcal{A}_1'(1)}\,\phi^{|n-i+1|}\Big]\sum_{j=i}^{+\infty} \ell_j\,, \quad 1 < i \neq n{+}1\,; \tag{101}$$

$$\mathrm{E}[m_{n+1}(J{+}n)] \;=\; 1 + \Big(\sum_{j=n+1}^{+\infty} \ell_j\Big)\,\frac{\mathcal{A}_1''(1)}{\mathcal{A}_1'(1)}\;. \tag{102}$$

Since the mean value of the total number of packet arrivals $\mathrm{E}[a(J{+}n)]$ is given by the sum of all $\mathrm{E}[m_i(J{+}n)]$, $i \geqslant 1$, we can now appropriately substitute the terms of this sum by the expressions (96), (101) and (102), depending on whether $i = 1$, $1 < i \neq n{+}1$ or $i = n{+}1$ respectively. This yields:

$$\begin{aligned}
&\mathrm{E}[a(J{+}n)]\\
&\;=\; 1 + \mathcal{A}_1'(1)L'(1) + \frac{\mathcal{A}_1''(1) - \big(\mathcal{A}_1'(1)\big)^2}{\mathcal{A}_1'(1)}\Big(\sum_{j=n+1}^{+\infty} \ell_j\Big) + \sigma(1{-}\sigma)\,\frac{(M_1'(1) - M_0'(1))^2}{\mathcal{A}_1'(1)}\\
&\qquad \cdot \frac{\phi}{1{-}\phi}\Big[- \phi^n + \sum_{j=n}^{+\infty} \ell_j + \sum_{j=n+2}^{+\infty} \ell_j + \sum_{j=1}^{n-1} \phi^{n-j}\,\ell_j - \sum_{j=n+2}^{+\infty} \phi^{j-n-1}\,\ell_j\Big]\,.
\end{aligned} \tag{103}$$

Finally, substitution of the results (90), (91) and (103) in (89) leads to an expression for the mean message delay $\mathrm{E}[c]$. After some very tedious calculations, we find

$$\begin{aligned}
\mathrm{E}[c] \;=&\; \mathrm{E}[u] + L'(1) + \frac{L'(1)}{2\,\mathcal{A}_1'(1)}\Big[\big(\mathcal{A}_1'(1)\big)^2(2L'(1) - 5) + 2\mathcal{A}_1''(1)\Big]\\
&+ \sigma(K{-}1)\frac{M_1'(1) - M_0'(1)}{\mathcal{A}_1'(1)}\Big[M_1'(1) - 1 + \mathcal{A}_1'(1)(L'(1) - 1)\\
&\qquad + (1{-}\sigma)\big(M_1'(1) - M_0'(1)\big)\Big(2L'(1) - 2 - (K{-}\frac{1}{2})\big(1 - \frac{L(\phi)}{\phi}\big)\Big)\Big]\\
&- \frac{\mathcal{A}_1''(1) - \big(\mathcal{A}_1'(1)\big)^2}{\mathcal{A}_1'(1)}\sum_{i=1}^{+\infty}\sum_{j=0}^{+\infty}(j{+}\frac{1}{2})\,\ell_i\,\ell_{i+j} \;+\; 2\,\sigma(1{-}\sigma)(K{-}1)\\
&\qquad \cdot \frac{\big(M_1'(1) - M_0'(1)\big)^2}{\mathcal{A}_1'(1)}\sum_{i=1}^{+\infty}\sum_{j=1}^{+\infty}\Big(K - j - \frac{(K{-}1/2)^2}{K}\,\phi^{j-1}\Big)\,\ell_i\,\ell_{i+j}\;.
\end{aligned} \tag{104}$$

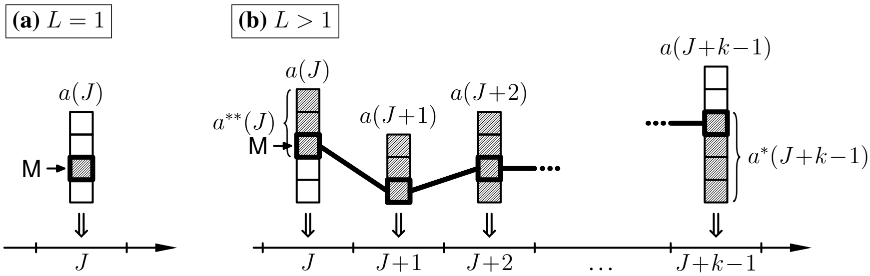


Figure 2.8: Arrival of a message M of length $L = k$. The packets that are transmitted during the transmission time of M are shaded.

2.5.2 Mean message transmission time

We define the *transmission time* of a message as the time period between the beginning of the slot in which the leading packet of the message is transmitted and the end of the slot in which the last packet is transmitted. Let h denote the transmission time of an arbitrary message during equilibrium, then we show in this section, how the previously obtained results can be used to compute the mean transmission time $\mathrm{E}[h]$. Specifically, let us consider again the tagged message M and condition on the length L of M, i.e.

$$\mathrm{E}[h] \;=\; \sum_{k=1}^{+\infty} \mathrm{E}[h_k]\,\ell_k\,, \tag{105}$$

where h_k denotes the transmission time of an arbitrary message of length k. Obviously, if $L = 1$, then $h_1 = 1$. On the other hand, if $L = k > 1$, then

$$h_k \;=\; a^{**}(J) + \sum_{n=1}^{k-2} a(J+n) + a^*(J+k-1) \quad , \; k > 1\,, \tag{106}$$

where $a^{**}(J)$ indicates the number of packets arriving in slot J, but to be transmitted *no sooner* than the packet of M that arrives in slot J (see Fig. 2.8). Note that, since the position of this packet among the arrivals in slot J is random, $a^{**}(J)$ has the same distribution as $a^*(J)$ and its mean value is given by (88). Thus, it follows from (105) and (106) that

$$\begin{aligned}
\mathrm{E}[h] \;&=\; 1 + \frac{1}{2}\,\mathrm{E}[a(J)](1-\ell_1) + \sum_{k=2}^{+\infty}\sum_{n=1}^{k-1} \mathrm{E}[a(J+n)]\,\ell_k - \frac{1}{2}\sum_{k=2}^{+\infty} \mathrm{E}[a(J+k-1)]\,\ell_k \\
&=\; \mathrm{E}[c] - \mathrm{E}[u(J)] + \frac{1}{2} + \frac{1}{2}\,\mathrm{E}[a(J)]\,,
\end{aligned} \tag{107}$$

where we have used (89). All terms appearing in (107) were studied in the previous section, and we arrive at the following closed-form expression for the mean message

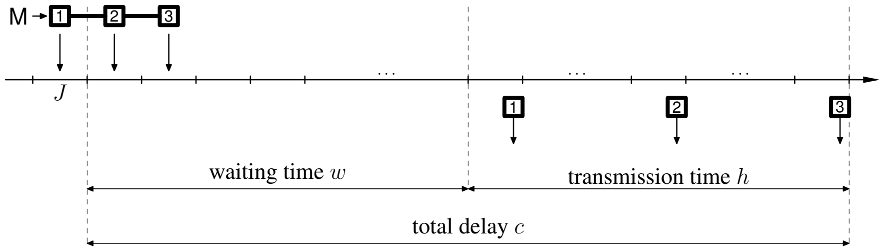


Figure 2.9: The waiting time w, transmission time h and total delay c of a message M of length 3, arriving in slot J.

transmission time $\mathrm{E}[h]$:

$$
\begin{aligned}
\mathrm{E}[h] \; &= L'(1) + \frac{1}{2}\,\mathcal{A}_1'(1) + \mathcal{A}_1'(1)L'(1)(L'(1)-2) + \frac{\mathcal{A}_1''(1)}{\mathcal{A}_1'(1)}\big(L'(1)-\frac{1}{2}\big) \qquad (108)\\
&\quad - \frac{\mathcal{A}_1''(1) - \big(\mathcal{A}_1'(1)\big)^2}{\mathcal{A}_1'(1)} \sum_{i=1}^{+\infty}\sum_{j=0}^{+\infty}(j+\frac{1}{2})\,\ell_i\,\ell_{i+j} \;+\; 2\,\sigma(1-\sigma)(K-1)\\
&\quad \cdot \frac{\big(M_1'(1)-M_0'(1)\big)^2}{\mathcal{A}_1'(1)}\Big[L'(1)-1-(K-\frac{1}{2})\big(1-\frac{L(\phi)}{\phi}\big)\\
&\qquad\qquad + \sum_{i=1}^{+\infty}\sum_{j=1}^{+\infty}\Big(K-j-\frac{(K-1/2)^2}{K}\,\phi^{j-1}\Big)\,\ell_i\,\ell_{i+j}\Big]\,.
\end{aligned}
$$

In addition to the message delay and transmission time, many authors (e.g. [47, 54, 207]) also study the *message waiting time*. We define the waiting time w of a message during equilibrium as the time period between the end of the slot during which the first packet of the message arrives and the time instant at which the transmission of this packet is about to start. From this definition and from Fig. 2.9, it is clear that $w = c - h$. Therefore, from (107), the mean waiting time $\mathrm{E}[w]$ is given by

$$
\begin{aligned}
\mathrm{E}[w] \;&=\; \mathrm{E}[u(J)] - \frac{1}{2} - \frac{1}{2}\,\mathrm{E}[a(J)]\\
&=\; \mathrm{E}[u] + \frac{\mathcal{A}_1''(1)}{2\,\mathcal{A}_1'(1)} - \frac{1}{2}\mathcal{A}_1'(1)(L'(1)+1) + \sigma\,(K-1)\,\frac{M_1'(1)-M_0'(1)}{\mathcal{A}_1'(1)}\Big[M_1'(1)-1\\
&\quad + \mathcal{A}_1'(1)(L'(1)-1) + (1-\sigma)(K-\frac{1}{2})\big(M_1'(1)-M_0'(1)\big)\big(1-\frac{L(\phi)}{\phi}\big)\Big]\,.
\end{aligned}
\qquad (109)
$$

2.5.3 Tail behaviour of the message delay

Whereas in Sec. 2.5.1, we have considered only the first moment of the message delay c, we now want to study the z-transform of its complete distribution and give an approximation for the distribution of the tail that can be implemented numerically. Specifically, we first construct an expression for the pgf $c_k(z)$ of c_k, the delay of an arbitrary message that consists of k packets. Obviously, the pgf of the unconditional message delay c is then given by

$$c(z) \;=\; \sum_{k=1}^{+\infty} c_k(z)\,\ell_k\,.$$

To assess the tail distribution of c_k, we use again the dominant pole approximation described in [43,46]. In other words, we approximate the probability for a message of length k to experience a delay of more than C slots by the following geometric form for sufficiently large values of C:

$$\text{Prob}[\,c_k > C\,] \;\cong\; -\frac{\theta^{c_k}}{z_0-1}\left(\frac{1}{z_0}\right)^{C+1}\,, \tag{110}$$

where θ^{c_k} is the residue of $c_k(z)$ in the point $z=z_0$. To identify z_0 and θ^{c_k} we thus need an expression for $c_k(z)$.

Let us consider again an arbitrary message M of L packets of which the first one arrives during slot J. We first treat the case where $\underline{L=k>1}$. Then it follows from (87) and (88) that

$$\begin{aligned}
c_k(z) \;&=\; \mathrm{E}\Big[z^{u(J)}\cdot z^{a(J+1)}\cdot z^{a(J+2)}\cdot\ldots\cdot z^{a(J+k-2)}\cdot z^{a^*(J+k-1)}\Big]\\
&=\; \frac{z}{z-1}\,\mathrm{E}\Big[z^{a(J+1)}\cdot z^{a(J+2)}\cdot\ldots\cdot z^{a(J+k-2)}\cdot\frac{z^{a(J+k-1)}-1}{a(J+k-1)}\cdot z^{u(J)}\Big]\\
&=\; \frac{z}{z-1}\int_1^z \frac{1}{y_k}\,\Delta_k(z,z,\ldots,z,y_k,z)\,\mathrm{d}y_k\,,
\end{aligned} \tag{111}$$

where we have introduced the joint generating function of the queue content at the end of slot J, together with the numbers of packet arrivals in each of the following $k-1$ slots:

$$\Delta_k(y_2,y_3,\ldots,y_k,z) \triangleq \mathrm{E}\Big[y_2^{a(J+1)}\cdot y_3^{a(J+2)}\cdot\ldots\cdot y_k^{a(J+k-1)}\cdot z^{u(J)}\Big]\,. \tag{112}$$

Now, following a similar line of thought as in Sec. 2.5.1, it is possible to relate the random variables in (112) to the system state variables for slot J only (this is demonstrated in Appendix 2.B in case all arguments of Δ_k are equal to z). We can therefore rely on result (133) of Appendix 2.A for the joint pgf $\widehat{P}$ so as to find an exact expression for $\Delta_k(z,z,\ldots,y_k,z)$ and hence, for $c_k(z)$. We have carried out the necessary calculations and found that the dominant pole z_0 is given implicitly by $z_0=G(z_0)$, where $G(z)$ is defined in (64). Its value can thus easily be obtained numerically by using e.g. the Newton-Raphson scheme. However, to find the tail behaviour, we also need θ^{c_k}, which follows from (111) as:

$$\theta^{c_k} \;=\; \operatorname*{Res}_{z=z_0}\,c_k(z) \;=\; \frac{z_0}{z_0-1}\int_1^{z_0}\frac{1}{y_k}\,\operatorname*{Res}_{z=z_0}\,\Delta_k(z,z,\ldots,z,y_k,z)\,\mathrm{d}y_k\,. \tag{113}$$

As such, the evaluation of θ^{c_k} involves the integration over a very complicated function of y_k. Indeed, as it turns out, the integrand of (113) is a product of an infinite number of factors in y_k, which makes it hard to implement the integration numerically. Similarly, for the case $\underline{L=1}$, it follows from (86) and (88) that

$$\begin{aligned} c_1(z) &= \mathrm{E}\Big[z^{u(J)} \cdot z^{-a(J)} \cdot z^{a^*(J)}\Big] \;=\; \frac{z}{z-1}\,\mathrm{E}\Big[\frac{1-z^{-a(J)}}{a(J)} \cdot z^{u(J)}\Big] \\ &= \frac{z}{z-1} \int_{z^{-1}}^{1} \frac{1}{y}\,\widehat{P}(1,y,y,\ldots,z)\,\mathrm{d}y\,. \end{aligned} \tag{114}$$

Here we can draw the same conclusions regarding z_0 and θ^{c_1} as above.

In short, although we might be able to give an analytically exact expression for $c_k(z)$ ($k \geqslant 1$) and its residue θ^{c_k} in z_0, the actual evaluation of the latter would be very cumbersome due to the integrations in (111) and (114). Note that this integral arises from the fact that the arrivals in the last arrival slot of message M are only partially comprised within c_k, i.e. it is the last term in (86) and (87) which gives difficulties. In the following paragraphs we try to avoid this problem by proposing an appropriate lower and upper bound for c_k.

An upper bound $\overline{c}_k$ for c_k.

Let us call $\overline{c}_k$ the delay of a message M with length k in the worst-case scenario that of all the arrivals in a certain slot, the packet of M is always the *last* one to be transmitted. This means that the term $a^*(J+n)$, $n \geqslant 0$, in (86) and (87) is always equal to its maximum value $a(J+n)$ and we have that

$$c_k \;\leqslant\; \overline{c}_k \;\triangleq\; u(J) + \sum_{n=1}^{k-1} a(J+n) \quad ,k \geqslant 1\,,$$

with generating function

$$\overline{c}_k(z) \;=\; \mathrm{E}\Big[z^{\sum_{n=1}^{k-1} a(J+n)} \cdot z^{u(J)}\Big] \;=\; \Delta_k(\underbrace{z, z, \ldots, z}_{k-1}, z)\,.$$

In Appendix 2.B we calculate the pgf $\Delta_k(z,z,\ldots,z,z)$ in terms of the joint pgf $\widehat{P}$, which in turn is calculated in Appendix 2.A. Using the obtained expressions (133) and (139) then yields an explicit expression for $\overline{c}_k(z)$ in terms of the parameters of the arrival process only. Again, we can show that the dominant pole z_0 of $\overline{c}_k(z)$ is given by $G(z_0) = z_0$. Its residue $\theta^{\overline{c}_k}$ in the point $z = z_0$ then follows as:

$$\begin{aligned} \theta^{\overline{c}_k} &= \Lambda_0(z_0)\,\Lambda_1(z_0)\,\Lambda_2(z_0)\cdot\ldots\cdot\Lambda_{k-2}(z_0)\,\frac{z_0^k}{\mathcal{A}_1'(1)}\,\frac{p_0(z_0-1)}{1-G'(z_0)} \\ &\cdot \left(\prod_{n=0}^{+\infty} \frac{1}{z_0} M_0(g_n^\circ)\,r_0(\gamma_n^\circ)\right)\left(\frac{z_0\,M_0'\big(z_0\,h_{1,k-1}(z_0)\big)}{M_0\big(z_0\,h_{1,k-1}(z_0)\big)} + \sum_{n=0}^{+\infty} \frac{1-\beta}{r_0(\gamma_n^\circ)} \cdot \frac{\partial}{\partial y_1}\gamma_n^\circ\right), \end{aligned} \tag{115}$$

where we make use of these particular evaluations of the functions (128) and (129) from Appendix 2.A:

- $\tilde{g}_n^\circ \;=\; \tilde{g}_n(z_0) \;=\; g_n(1, z_0)\,, \qquad n \geqslant 0\,;$ (116)
- $\begin{cases} \tilde{\gamma}_0^\circ \;=\; \tilde{\gamma}_0(z_0) \;=\; \gamma_0(1,1,z_0) \\ \tilde{\gamma}_n^\circ \;=\; Q(\tilde{\gamma}_{n-1}^\circ)\,\mu(\tilde{g}_n^\circ) \quad ,n > 0\,; \end{cases}$ (117)
- $g_n^\circ \;=\; g_n\big(h_{n+1,k-1}(z_0), z_0\big) \;=\; \sum_{j=1}^{n} \ell_j\, z_0^j \;+\; z_0^{n+1}\, h_{n+1,k-1}(z_0) \sum_{j=n+1}^{+\infty} \ell_j\,;$ (118)
- $\begin{cases} \gamma_0^\circ \;=\; \gamma_0\big(\,Q\big(\tilde{\gamma}_{k-2}^\circ\big)\,,\, h_{1,k-1}(z_0)\,,\, z_0\,\big) \;=\; Q\big(\tilde{\gamma}_{k-2}^\circ\big)\,\mu\big(z_0\, h_{1,k-1}(z_0)\big) \\ \gamma_n^\circ \;=\; \gamma_n\big(\,Q\big(\tilde{\gamma}_{k-2}^\circ\big)\,,\, h_{1,k-1}(z_0)\,,\, h_{2,k-1}(z_0)\,,\,\ldots\,,\, h_{n+1,k-1}(z_0)\,,\, z_0\,\big) \\ \qquad = Q(\gamma_{n-1}^\circ)\,\mu(g_n^\circ) \quad ,n > 0\,; \end{cases}$ (119)
- $\begin{cases} \dfrac{\partial}{\partial y_1}\gamma_0^\circ \;=\; Q\big(\tilde{\gamma}_{k-2}^\circ\big)\,\mu'\big(z_0\, h_{1,k-1}(z_0)\big)\, z_0 \\ \dfrac{\partial}{\partial y_1}\gamma_n^\circ \;=\; Q'(\gamma_{n-1}^\circ)\,\Big(\dfrac{\partial}{\partial y_1}\gamma_{n-1}^\circ\Big)\,\mu(g_n^\circ) \quad ,n > 0\,. \end{cases}$ (120)

In (115)–(120) we also make use of the functions $h_{ij}(z)$ $(i, j \geqslant 0)$, $\Lambda_k(z)$ $(k \geqslant 0)$ defined in Appendix 2.B, by (136) and (138) respectively. For large n, the factors of the infinite product in (115) converge to $G(z_0)/z_0 = 1$. Likewise, one can prove the infinite series to be also convergent. Therefore, it is not too difficult to evaluate $\theta^{\overline{c}_k}$ numerically based on the expressions (115)–(120).

A lower bound $\underline{c}_k$ for c_k.

We can also consider the scenario in which the packets of M always get transmitted *first*, i.e. the term $a^*(J{+}n)$, $n \geqslant 0$ in (86) and (87) always equals its minimum value 1. Under this condition, the message delay $\underline{c}_k$ is given by

$$c_k \;\geqslant\; \underline{c}_k \;\triangleq\; \begin{cases} u(J) + \displaystyle\sum_{n=1}^{k-2} a(J{+}n) + 1 & ,k \geqslant 2\,; \\ u(J) - a(J) + 1 & ,k = 1\,. \end{cases}$$

In the case where $\underline{L = k > 1}$, we can remark that $\underline{c}_k \;=\; \overline{c}_{k-1} + 1$. Hence, the pgf of $\underline{c}_k$ is given by $\underline{c}_k(z) \,=\, z\,\overline{c}_{k-1}(z)$ and has the same dominant pole z_0 as $\overline{c}_{k-1}(z)$. Its residue in the point $z{=}z_0$ is then simply given by

$$\theta^{\underline{c}_k} \;=\; z_0\,\theta^{\overline{c}_{k-1}} \quad ,k \geqslant 2\,,$$

so we can reuse the result of (115). In case $\underline{L = 1}$, we find for $\underline{c}_1(z)$:

$$\underline{c}_1(z) \;=\; \mathrm{E}[z^{u(J)} \cdot z^{-a(J)} \cdot z] \;=\; z\;\widehat{P}\Big(1\,,\, \frac{1}{z}\,,\, \frac{1}{z}\,,\, \frac{1}{z}\,,\,\ldots\,,\, z\Big)\,.$$

As such, it is possible to devise a similar algorithm for $\theta^{\underline{c}_1}$ as in (115)–(120), albeit much simpler here.

To conclude, we note that once the values z_0, $\theta^{\overline{c}_k}$ and $\theta^{\underline{c}_k}$, $k \geqslant 1$, are obtained, one can calculate from (110) the following upper and lower bounds for the probability of the *unconditional* message delay c to be larger than C:

$$-\frac{\theta^{\underline{c}}}{z_0-1}\, z_0^{-C-1} \leqslant \text{Prob}[c > C] \leqslant -\frac{\theta^{\overline{c}}}{z_0-1}\, z_0^{-C-1}\,,$$

where $\theta^{\underline{c}} = \sum_{k=1}^{+\infty} \theta^{\underline{c}_k}\, \ell_k$ and $\theta^{\overline{c}} = \sum_{k=1}^{+\infty} \theta^{\overline{c}_k}\, \ell_k$. In practice, these bounds prove to be reasonably tight, as will be demonstrated by the examples in the next section.

2.6 Numerical examples and discussion of results

2.6.1 Effect of primary and secondary correlation on the mean queue content

As mentioned before, the correlated train arrivals model for the packet arrival process considered in this chapter exhibits a twofold correlation: a primary correlation due to the arrival of messages in trains at the rate of one packet per slot and a secondary correlation due to the nonindependent generation of new messages. In this section, we will demonstrate the importance of taking into account both types of correlation. For this purpose, we compare the results obtained for the mean queue content of the '*correlated train arrivals*' model with the results that would be found if a model without secondary correlation or an uncorrelated model for the packet arrival process were used.

For the model without secondary correlation but with primary correlation ('*uncorrelated train arrivals*' model), we assume that new messages are generated by the user population independently from slot to slot according to the pgf $\mathcal{A}_1(z)$ given by equation (53); the message lengths are iid with pgf $L(z)$. In this case, the mean equilibrium queue content $\text{E}[u_{\text{prim}}]$ is given by [207]

$$\text{E}[u_{\text{prim}}] = \mathcal{A}_1'(1)L'(1) + \left[\frac{\mathcal{A}_1''(1)L'(1) + \mathcal{A}_1'(1)^2 L''(1)}{1-\rho}\right]\frac{L'(1)}{2}\,. \tag{121}$$

By using (53), we can express $\text{E}[u_{\text{prim}}]$ in terms of the basic parameters of our correlated train arrivals model as

$$\begin{aligned}\text{E}[u_{\text{prim}}] &= \frac{\mathcal{A}_1'(1)^2 L'(1)}{2(1-\rho)} \cdot \text{Var}[\ell] \\ &\quad + \frac{\sigma L'(1)^2}{2(1-\rho)} \cdot \text{Var}[w^{(1)}] + \frac{(1-\sigma)L'(1)^2}{2(1-\rho)} \cdot \text{Var}[w^{(0)}] \\ &\quad + \frac{L'(1)}{2}\left[\mathcal{A}_1'(1)\big(2-L'(1)\big) + \sigma(1-\sigma)\frac{\big[M_1'(1)-M_0'(1)\big]^2 L'(1)}{1-\rho}\right].\end{aligned} \tag{122}$$

Comparing the expressions (122) and (66) for $\text{E}[u_{\text{prim}}]$ and $\text{E}[u]$, we find

$$\text{E}[u] = \text{E}[u_{\text{prim}}] + \frac{\sigma(M_1'(1)-M_0'(1))(M_1'(1)L'(1)-1)L'(1)}{1-\rho} \cdot (K-1)\,. \tag{123}$$

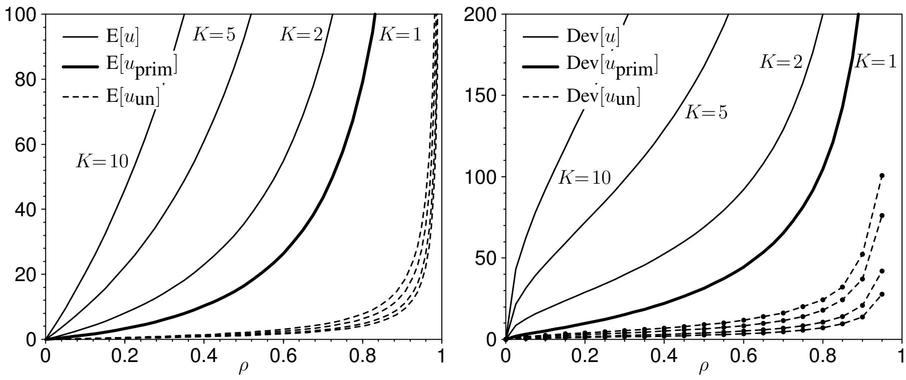


Figure 2.10: Expected value and standard deviation of the queue content versus the total load ρ for various values of K. The message-length distribution is a mixture of two geometrics ($p = 0.5$, mean 10 and variance 150), $M_0(z) = 1$ and $M_1(z) = 0.5z/(1 - 0.5z)$.

This clearly shows that for given $L(z)$, $M_0(z)$, $M_1(z)$ and $\sigma \neq 0$, $\mathrm{E}[u]$ is always greater than $\mathrm{E}[u_{\text{prim}}]$ when $K > 1$, i.e., in case of a positive correlation coefficient $\phi = 1 - 1/K$ between the environment states in two consecutive slots; the expressions only agree when $K = 1$ ($\phi = 0$), i.e., in case of an uncorrelated message generation process. Therefore, neglecting the (positive) correlation in the message generation always leads to an underestimation of the mean queue content.

In the '*uncorrelated*' model, packets arrive to the multiplexer independently from slot to slot. In order to make a fair comparison we assume the distribution of the number of packet arrivals per slot to be the same as in the correlated train arrivals model, i.e. the mean and variance are equal to ρ and $\mathrm{Var}[a]$, given in equations (58) and (80). The mean equilibrium queue content $\mathrm{E}[u_{\text{un}}]$ in the uncorrelated model is then given by [56, 102]

$$\mathrm{E}[u_{\text{un}}] = \frac{\rho}{2} + \frac{\mathrm{Var}[a]}{2(1-\rho)} \,. \tag{124}$$

We now consider some practical examples. First, let us introduce four possible choices for the pgf $L(z)$ of the message length ℓ :

$$L_1(z) = z^m \,, \qquad L_2(z) = \frac{(1-\theta)^2 z}{(1-\theta\, z)^2} \,,$$
$$L_3(z) = \frac{(1-\lambda)z}{1-\lambda\, z} \,, \qquad L_4(z) = p\,\frac{(1-\lambda_1)z}{1-\lambda_1\, z} + (1-p)\,\frac{(1-\lambda_2)z}{1-\lambda_2\, z} \,, \tag{125}$$

i.e., fixed-length messages, a negative binomial distribution, a geometric distribution and a mixture of two geometric distributions, respectively. The parameters of the distributions are chosen such that the mean message length $L'(1)$ is equal to a given value m. Additionally, in case of $L_4(z)$, a value for p and $\mathrm{Var}[\ell_4]$ must be specified. The variances of the other message-length distributions are given by

$$\mathrm{Var}[\ell_1] = 0 \,, \qquad \mathrm{Var}[\ell_2] = \frac{1}{2}(m-1)\,(m+1) \,, \qquad \mathrm{Var}[\ell_3] = m(m-1) \,. \tag{126}$$

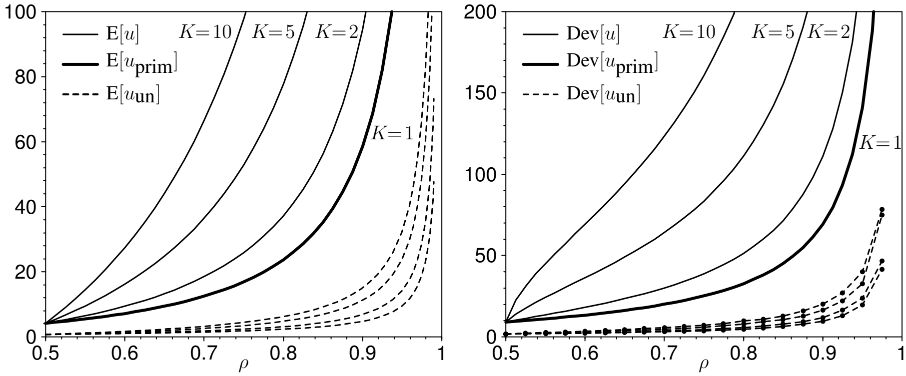


Figure 2.11: Expected value and standard deviation of the queue content versus the total load ρ for various values of K. The message-length distribution is negative binomial (mean 10), $M_0(z)=\exp[0.05(z-1)]$ and $M_1(z)=z$.

In Figs. 2.10, 2.11 and 2.12**(a)**, the mean E[$\cdot$] and standard deviation Dev[$\cdot$] of the queue content for u (correlated train arrivals), u_{prim} (uncorrelated train arrivals) and u_{un} (uncorrelated packet arrivals), are plotted versus the total load $\rho = \big(\sigma M_1'(1) + (1-\sigma)M_0'(1)\big)L'(1)$. For each of the curves in these plots, a particular fixed choice is made for the distributions of ℓ, $w^{(1)}$ and $w^{(0)}$ and for the value of the correlation factor K. The variation of the load along the horizontal axis is brought about only by varying σ between 0 and a maximum value implied by the stability condition. Hence, ρ can range from $M_0'(1)L'(1)$ to 1.

In Figs. 2.10 and 2.11, E[u], E[u_{prim}] and E[u_{un}], as well as Dev[u], Dev[u_{prim}] and Dev[u_{un}] are plotted for different values of the environment correlation factor K, namely $K = 1, 2, 5, 10$. In Fig. 2.10, the message-length distribution is a mixture of two geometrics according to $L_4(z)$ with $p = 0.5$, $m = 10$ and a variance of 150. The number of new messages in the 1-slots is assumed to have a geometric distribution with an expected value of 2, whereas no new messages are generated during the 0-slots. In Fig. 2.11, we give an example in which new messages are generated during 0-slots too: $M_0(z) = \exp[q(z-1)]$, i.e., a Poisson distribution with intensity $q = 0.05$. Furthermore, exactly one new message starts per 1-slot ($M_1(z) = z$) and the message lengths are negative binomially distributed with $m = 10$. Note that in this case, the load ρ cannot become less than $mq = 0.5$ by merely varying σ. The figures clearly demonstrate the severe underestimation of the system performance when the different levels of correlation in the arrival process are neglected. For instance, we observe the rapid growth of E[u] as K increases, i.e., as the absolute (mean) lengths of the 1-periods and the 0-periods increase, even though the overall ratio σ of 1-slots remains unchanged. As is expected from (122) and (123), all the curves for E[u_{prim}] coincide with the one representing E[u] for $K = 1$ (bold curve). For Dev[u] and Dev[u_{prim}], similar conclusions can be drawn. The dashed curves represent E[u_{un}] and Dev[u_{un}], showing what happens if the correlation in the packet arrival stream is neglected altogether. We see that the moments of u_{un} also slightly increase with higher values of K, although not in the same drastic way as the moments of u. It should

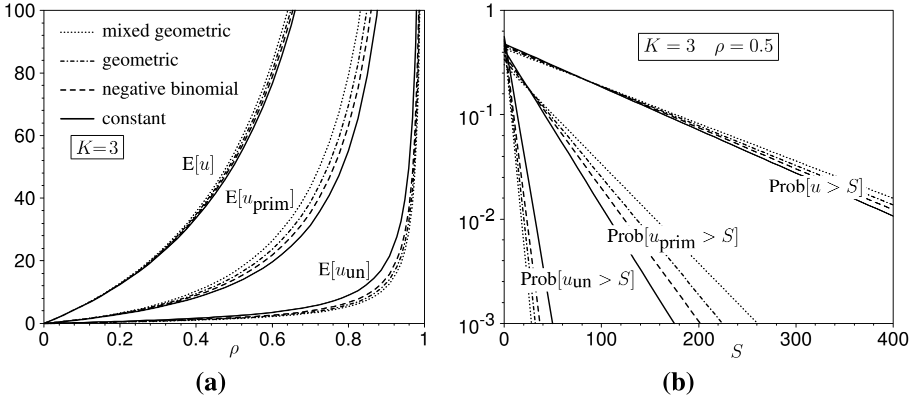


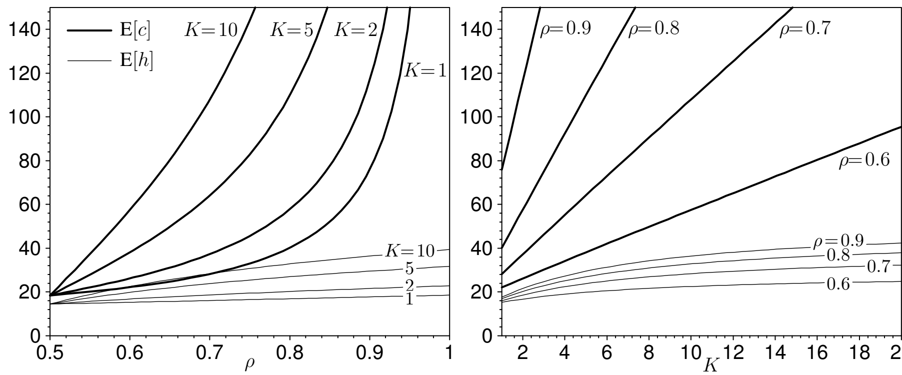
Figure 2.12: Mean queue content versus the total load ρ for $K=3$ in **(a)** and its tail distribution Prob$[\, u > S\,]$ against S for $K=3$ and a total load $\rho=0.5$ in **(b)**. These are plotted for various distributions of the message lengths (mean 10), $M_0(z)=1$ and $M_1(z)=0.5z/(1-0.5z)$.

be noted that in both Figs. 2.10 and 2.11, the results for Dev$[u_{\text{un}}]$ were obtained by simulation of the arrival process over a large number of slots and are included for comparison only.

In Fig. 2.12, the environment correlation factor is chosen to be $K=3$ for all curves. In the 1-slots, the number of new messages has a geometric distribution with an expected value of 2, while no new messages are generated during the 0-slots. To illustrate the influence of the distribution of the message lengths, we plotted the mean queue content and its tail distribution for the four pgfs given in (125). Their parameters are chosen such that $L'(1) = 10$, and for the mixed geometric distribution : $p = 0.5$, Var$[l_4] = 150$. In Fig. 2.12**(a)**, we know from (66) and (122) that for E$[u]$ and E$[u_{\text{prim}}]$ the difference between the four curves is due only to the linear impact of the message-length variance, which here has the values : Var$[\ell_1]=0$, Var$[\ell_2]=49.5$, Var$[\ell_3]=90$ and Var$[\ell_4]=150$. For E$[u_{\text{un}}]$, however, we see that the curves are ordered in the opposite way as for the correlated arrivals models. In view of (124), this suggests a negative dependence of the variance (80) of the number of packet arrivals per slot on the variance of the message lengths. In Fig. 2.12**(b)** we show a logarithmic plot of the tail distribution of the queue content. The probability for the queue content to exceed a certain level S is plotted for the same parameters of the arrival process as in Fig. 2.12**(a)** and for a load $\rho = 0.5$. Here, similar conclusions can be drawn and, even for this medium load, the difference between the curves for the three arrival models is particularly striking.

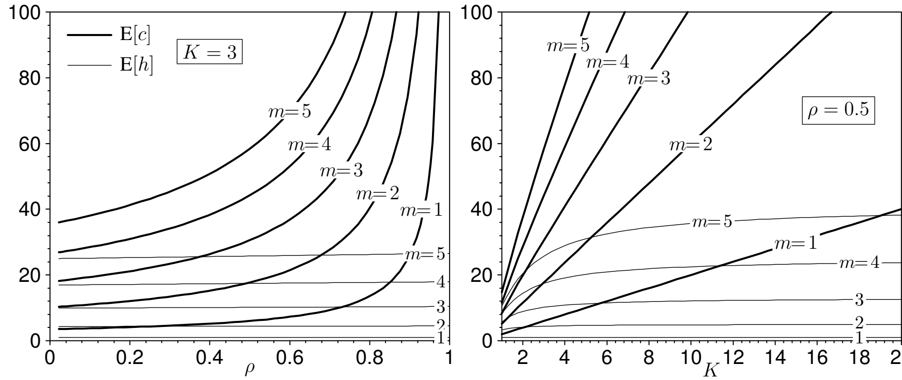
2.6.2 The message delay and transmission time

In order to illustrate how E$[c]$, E$[h]$ and Prob$[c > C]$ are influenced by the parameters of the arrival process, we now consider some practical examples. The Figs. 2.13, 2.14 and 2.15 all consist of two plots. In part **(a)** of these figures, the mean delay E$[c]$ and the mean transmission time E$[h]$ of the messages are plotted versus the total load



(a) **(b)**

Figure 2.13: Mean message delay $\mathrm{E}[c]$ and transmission time $\mathrm{E}[h]$ in case of a negative binomial message-length distribution (mean 10), $M_0(z) = e^{0.05(z-1)}$ and $M_1(z) = z$. In **(a)** these measures are plotted versus the total load ρ for various values of K, whereas in **(b)** they are plotted versus K, for different values of the load.



(a) **(b)**

Figure 2.14: Mean message delay $\mathrm{E}[c]$ and transmission time $\mathrm{E}[h]$ in case of a deterministic message length $L = m$, $M_0(z) = 1$ and $M_1(z) = z^2$, for various values of L. In **(a)** these are plotted versus the load ρ for $K = 3$ and in **(b)** versus K for $\rho = 0.5$.

$\rho = [\sigma M_1'(1) + (1-\sigma)M_0'(1)]L'(1)$ of the system. The variation of ρ along the abscissa is brought about only by varying σ between 0 and a maximum value implied by the stability condition. Hence, ρ can range from $M_0'(1)L'(1)$ to 1. In part **(b)** of each figure, $\mathrm{E}[c]$ and $\mathrm{E}[h]$ are plotted versus the burst-length factor K, for a particular fixed choice of the load (or equivalently, of σ).

In Fig. 2.13, the message-length distribution is negative binomial according to $L_3(z)$, with $m = 10$. The number of new messages in the 0-slots is assumed to have a Poisson distribution with intensity $q = 0.05$, whereas exactly one new message is generated in each 1-slot, i.e. $M_0(z) = e^{q(z-1)}$ and $M_1 = z$. Note that in this case, the load cannot be-

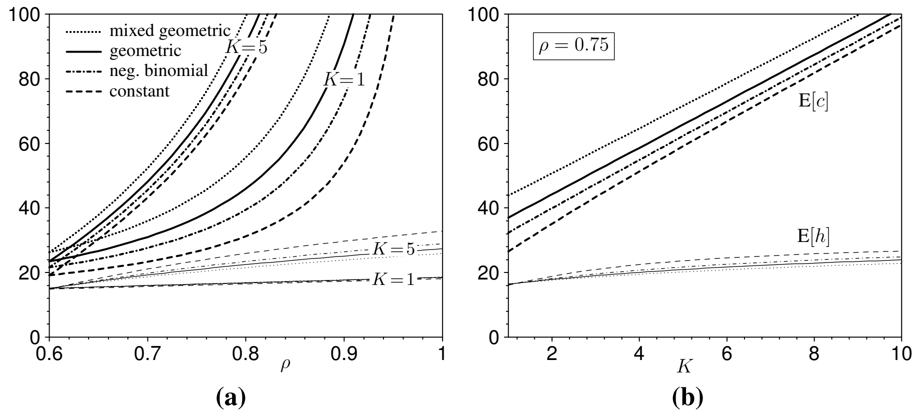
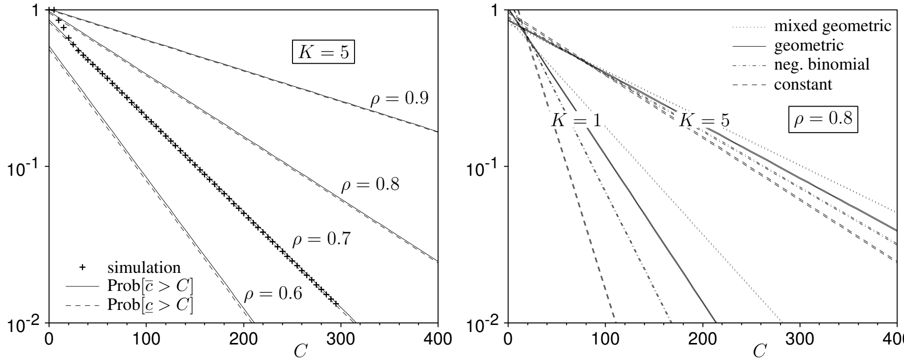


Figure 2.15: Mean message delay $\mathrm{E}[c]$ and transmission time $\mathrm{E}[h]$ for various distributions of the message length L (mean 10), $M_0(z) = 0.06z + 0.94$ and $M_1(z) = z$. In **(a)** these are plotted versus the load ρ for $K = 1, 5$ and in **(b)** versus K for $\rho = 0.75$.

come less than $mq = 0.5$ by merely varying σ. The first thing we notice on these plots, is the rapid growth of the mean message delay as the burst-length factor K increases, even though the load is unchanged. Specifically, Fig. 2.13**(b)** shows that this growth is almost linear, especially for large K. Therefore, we can conclude that in the analysis of multiplexers like the one studied here, it is very important to take into account possible correlation in the message generation process. In fact, suppose we would choose to neglect this secondary correlation and analyse the system under the assumption that the numbers of messages generated in each slot are no longer correlated, but are iid variables with pgf $\mathcal{A}_1(z)$. Of course, this pgf would then be: $\sigma M_1(z) + (1-\sigma) M_0(z)$. It can be seen that the results of the analysis for this model with only primary correlation can be obtained from our model (with both primary and secondary correlation) by assuming that the correlation coefficient ϕ of the environment state in two consecutive slots is zero, i.e. $K = 1$. Indeed, if $K = 1$ (and $\phi = 0$), our results (104) and (109) for $\mathrm{E}[c]$ and $\mathrm{E}[w]$ reduce to the ones obtained in [207], where an uncorrelated message generation process was considered. The next observation we make from Fig. 2.13 is that the influence of the environment parameters K and σ (or ρ) on the mean transmission time $\mathrm{E}[h]$ is rather limited (as opposed to the very significant influence on $\mathrm{E}[c]$). For instance, in Fig. 2.13**(a)** we see that, for given K, $\mathrm{E}[h]$ increases only slightly with the load ρ and remains bounded, even if the load is maximal. Indeed, during the transmission time of a message M, the only packets that are transmitted, are those that arrived during the L arrival slots of M. On the average, the amount of those packets is bounded for all $0 \leqslant \sigma \leqslant 1$, even for values of σ for which $\rho > 1$. Additionally, in Fig. 2.13**(b)**, we observe that, for given ρ, $\mathrm{E}[h]$ reaches a limit value if $K \to \infty$. Naturally, all of the above considerations are valid for the Figs. 2.14 and 2.15 too.

In Fig. 2.14, the message length is deterministic according to $L_1(z)$, and we have plotted $\mathrm{E}[c]$ and $\mathrm{E}[h]$ for $m = 1, 2, 3, 4, 5$. In the 0-slots, no new messages are generated, whereas in each 1-slot, there are exactly 2 new messages, i.e. $M_0(z) = 1$ and $M_1 = z^2$. In Fig. 2.14**(a)** and **(b)** we have assumed $K = 3$ and $\rho = 0.5$ respectively.



(a) **(b)**

Figure 2.16: Upper and lower bounds for Prob$[c > C]$ in case $M_0(z) = e^{0.05(z-1)}$ and $M_1(z) = z$. In **(a)** the tails of $\underline{c}$ and $\overline{c}$ are plotted for $K = 5$ and various values of the load. The messages have a constant length of 10 packets and for $\rho = 0.7$, a simulation result of c is added. In **(b)** we considered various distributions of the message length (mean 10), for $\rho = 0.8$ and $K = 1, 5$.

We observe that, for a given K and σ, both E$[c]$ and E$[h]$ increase with the message length m, as is intuitively clear.

In Fig. 2.15, in each 1-slot exactly one new message is generated, while in each 0-slot this only happens with probability 0.06, i.e. $M_0(z) = 0.06z + 0.94$ and $M_1(z) = z$. To illustrate the influence of the distribution of the message lengths, we have plotted E$[c]$ and E$[h]$ for the four pgfs given in (125). Their parameters are chosen such that $L'(1) = 10$, and Var$[L_4] = 150$. The other variances are then given by Var$[L_1] = 0$, Var$[L_2] = 49.5$ and Var$[L_3] = 90$. In Fig. 2.15**(a)** and 2.15**(b)** we chose $K = 1, 5$ and $\rho = 0.75$ respectively. Although it cannot directly be proven from (104), we make the empirical observation that the mean message delay E$[c]$ appears to increase with Var$[L]$ (we *did* prove that this is certainly the case for E$[u]$). However, for the mean transmission time E$[h]$, quite the opposite seems to hold: message-length distributions with higher variances yield a lower, hence better, mean transmission time.

Finally, in Fig. 2.16, we show two plots of the tail distribution of the message delay c, with $M_0(z) = e^{0.05(z-1)}$ and $M_1(z) = z$. First, in Fig. 2.16**(a)**, the messages are exactly 10 packets long. Both Prob$[\overline{c} > C]$ and Prob$[\underline{c} > C]$ are plotted for $K = 5$ and various values of the load ρ. We also used these parameters to run a simulation according to the mathematical model described in Sec. 2.2. The delay distribution of the messages generated during a $2 \cdot 10^8$-slot simulation for $\rho = 0.7$ is added to the figure, matching exactly the result predicted in Sec. 2.5.3. Secondly, in Fig. 2.16**(b)**, the upper and lower bounds for Prob$[c > C]$ are plotted for the same distributions of the message length as in Fig. 2.15, for $\rho = 0.8$ and $K = 1, 5$. Again, we see that the message-length distribution with the highest variance gives the highest tail probabilities for the delay. Both figures show that the upper and lower bounds for the tail distribution of c which we obtained in Sec. 2.5.3 lie extremely close together.

The influence of the parameters of the model as observed in the above examples, is in accordance with the nowadays well established rule that queue performance worsens as the autocorrelation function of the arrival stream extends to larger time scales. With regard to our particular model, it would be useful to be able to quantify the 'amount' of correlation that can be seen as primary correlation (messages arrive as trains) and the amount qualified as secondary correlation (nonindependent environment) apart from each other, as well as their respective impact on the multiplexer performance. For the secondary correlation, this is not too difficult. It is clear from (35) that the autocorrelation of the environment $\{r_k\}$ at lag n behaves like ϕ^n, so a higher ϕ (or K) means a higher secondary correlation. Also, the impact of K on the performance measures is clearly discussed above. A similar characterisation for the primary correlation is not so easy to provide. A possible candidate could be the mean message length $L'(1)$, since it reveals how long an average message affects the arrival stream. Unfortunately, unlike K, $L'(1)$ also has influence on the first-order statistics of the arrival stream, which makes it difficult to use this parameter to quantify correlation.

2.7 Conclusion

In this chapter, we have analyzed a discrete-time queueing system with correlated variable-length packet-train arrivals that represents a statistical multiplexer with an unbounded population of users. The simultaneous impact of *two* types of correlation in the packet arrival process has been assessed, a primary correlation due to the fact that the messages generated by the users enter the system at the rate of one packet per slot (train arrivals) and a secondary correlation due to the fact that the number of messages generated each slot depends on a Markov-modulated environment.

The use of the DSVT technique with an infinite-dimensional state description has allowed us to obtain expressions for most of the relevant performance measures, both in terms of packets and in terms of messages. For the packets, we have explicit expressions for the moments of the queue content and the packet delay during equilibrium. An algorithmic solution for the tail distribution is given as well. In terms of messages, we obtained the mean value of the message delay and the message transmission time, where the latter is the time from the instant the first packet of a message starts its service until the last packet of the message leaves the queue. For both the delay and transmission time of the messages, we have a tight upper and lower bound for their tail distribution. Our results convincingly show that the required buffer storage capacity for a statistical multiplexer may be severely underestimated if the correlation in the packet arrival process is not taken into account properly. Correspondingly, the delay of both packets and messages is equally influenced by the correlation in the arrival stream.

2.A Appendix: An iterative solution for the functional equation

We now derive an explicit formula for the joint pgf of system state variables, observed both on arbitrary slot boundaries (P) and at the start of new messages ($\widehat{P}$). To this end, we repeatedly apply functional equation (49) as follows:

$$
\begin{aligned}
&P(x, y_1, y_2, \ldots, z) \\
&\quad = \frac{1}{z}\, M_0(y_1 z)\, r_0\big(x\,\mu(y_1 z)\big) \\
&\qquad \cdot \Big[p_0\,(z-1) + P\,\big(Q\big(x\,\mu(y_1 z)\big), D_1(y_2 z), D_2(y_3 z), \ldots, z\big)\Big] \\
&\quad = \frac{1}{z}\, M_0(y_1 z)\, r_0\big(x\,\mu(y_1 z)\big)\, p_0\,(z-1) \\
&\qquad + \frac{1}{z}\, M_0(y_1 z)\, r_0\big(x\,\mu(y_1 z)\big)\, \frac{1}{z}\, M_0\big(z D_1(y_2 z)\big)\, r_0\big(Q(x\,\mu(y_1 z))\,\mu(z D_1(y_2 z))\big) \\
&\qquad\quad \cdot \Big[p_0\,(z-1) + P\Big(Q\big(Q(x\,\mu(y_1 z))\,\mu(z D_1(y_2 z))\big), \\
&\qquad\qquad\qquad\qquad D_1(z D_2(y_3 z)), D_2(z D_3(y_4 z)), \ldots, z\Big)\Big] \\
&\quad = \ \ldots \\
&\quad = p_0(z-1) \sum_{j=0}^{+\infty} \left(\prod_{n=0}^{j} \frac{1}{z} M_0(g_n)\, r_0(\gamma_n) \right) \\
&\qquad + \left(\prod_{n=0}^{+\infty} \frac{1}{z} M_0(g_n)\, r_0(\gamma_n) \right) \cdot \lim_{N\to\infty} P\big(\, Q(\gamma_N), D_1(\ldots z D_N(y_{N+1} z)\ldots), \\
&\qquad\qquad\qquad\qquad D_2(\ldots z D_{N+1}(y_{N+2} z)\ldots),\ \ldots\ , z\big)\,.
\end{aligned}
\tag{127}
$$

In this expression, we defined the functions g_n, $n \geqslant 0$, as

$$
\begin{aligned}
g_n \;=\; g_n(y_{n+1}, z) \;&\triangleq\; z\, D_1(z\, D_2(\ldots z\, D_n(y_{n+1} z)\ldots)) \\
&=\; \sum_{j=1}^{n} \ell_j\, z^j + y_{n+1}\, z^{n+1} \sum_{j=n+1}^{+\infty} \ell_j\,,
\end{aligned}
\tag{128}
$$

i.e. g_n is a polynomial of degree $n+1$ in z, converging to $L(z)$ as n goes to infinity. Secondly, the functions γ_n, $n \geqslant 0$, in (127) are defined recursively as

$$
\begin{cases}
\gamma_0 \;=\; \gamma_0(x, y_1, z) \;\triangleq\; x\,\mu(y_1 z)\,; \\
\gamma_n \;=\; \gamma_n(x, y_1, y_2, \ldots, y_{n+1}, z) \;\triangleq\; Q\big(\gamma_{n-1}\big)\,\mu\big(g_n\big) \quad , n \geqslant 1\,.
\end{cases}
\tag{129}
$$

A remarkable thing about this iterative procedure is that the arguments of P on the right-hand side of (127) become independent of all variables other than z, but also seem to converge to the one-dimensional trajectory $\big(\chi(z), \eta_1(z), \eta_2(z), \ldots, z\big)$ for which (63) holds. For the variables y_n, $n \geqslant 1$, this follows directly from (61). On the other hand, to prove that $Q(\gamma_\infty) = \chi(z)$, we first remark that if the sequence γ_n

converges to some limiting function γ_∞, then definition (129) implies that

$$Q(\gamma_\infty) = Q\big(Q(\gamma_\infty)\,\mu(L(z))\big); \qquad Q\big(\gamma_\infty(1,1,\ldots,1,1)\big) = 1\,, \tag{130}$$

since $\lim_{n\to\infty} g_n = L(z)$. Now, when comparing (62) and (130), we see that both $\chi(z)$ and $Q(\gamma_\infty)$ are implicitly determined by the same relations. Hence, we can conclude on their equality. Note also that a fortiori the 'reduced' sequence $\tilde{\gamma}_n(z)$ defined in (73) converges to $\chi(z)$ as well, so $\gamma_\infty = \tilde{\gamma}_\infty(z) = \chi(z)$.

We can also give a more 'constructive' proof for the convergence of $Q\big(\gamma_n(x, y_1,\ldots,y_{n+1},z)\big)$ to $\chi(z)$, in the case where the message length is bounded by N (see Appendix 2.C). From (128) and (129), we then have that $\gamma_n = Q(\gamma_{n-1})\,\mu(L(z))$, for all $n \geqslant N$. Therefore, the functions γ_n, $n \geqslant N$, depend only on the variables $(x, y_1, \ldots, y_N, z)$. Moreover, for a particular value of z, the dynamic behaviour of the sequence $\gamma_{n\geqslant N}$ is determined by the following linear fractional transformation (or Moebius-transform):

$$\gamma_n = \frac{\alpha\,\gamma_{n-1} + 1-\alpha}{(1-\beta)\,\gamma_{n-1} + \beta} \cdot \mu(L(z))\,. \tag{131}$$

This well-known type of transformations generally has one repelling and one attracting fixed point, the latter here being equal to $\chi(z)\,\mu(L(z))$. This means that, whatever the starting value γ_{N-1}, eventually the sequence $\{Q(\gamma_n)\}$ will always converge to the value $\chi(z)$.

Finally, following the above discussion, we can substitute the linear solution (63) of the functional equation into (127), which yields an explicit expression for the joint pgf of the system state variables in an arbitrary slot:

$$\begin{aligned} P(x, y_1, y_2, \ldots, z) &= p_0(z-1) \sum_{j=0}^{+\infty} \left(\prod_{n=0}^{j} \frac{1}{z} M_0(g_n)\, r_0(\gamma_n) \right) \\ &\quad + \frac{p_0(z-1)G(z)}{z - G(z)} \left(\prod_{n=0}^{+\infty} \frac{1}{z} M_0(g_n)\, r_0(\gamma_n) \right) . \end{aligned} \tag{132}$$

To obtain also the distribution as seen by an arbitrary new message, we need to differentiate this expression to y_1, as indicated by (84). Here it is useful to remark that g_n, $n \geqslant 1$, is independent of y_1. We obtain:

$$\begin{aligned} \widehat{P}(x, y_1, y_2, \ldots, z) &= \frac{y_1}{\mathcal{A}_1'(1)}\, p_0(z-1) \\ &\cdot \Big[\sum_{j=0}^{+\infty} \left(\prod_{n=0}^{j} \frac{1}{z} M_0(g_n)\, r_0(\gamma_n) \right) \left(\frac{z\, M_0'(y_1 z)}{M_0(y_1 z)} + \sum_{n=0}^{j} \frac{1-\beta}{r_0(\gamma_n)} \cdot \frac{\partial}{\partial y_1} \gamma_n \right) \\ &\quad + \frac{G(z)}{z - G(z)} \left(\prod_{n=0}^{+\infty} \frac{1}{z} M_0(g_n)\, r_0(\gamma_n) \right) \left(\frac{z\, M_0'(y_1 z)}{M_0(y_1 z)} + \sum_{n=0}^{+\infty} \frac{1-\beta}{r_0(\gamma_n)} \cdot \frac{\partial}{\partial y_1} \gamma_n \right) \Big] . \end{aligned} \tag{133}$$

Note that since $g_n \to L(z)$ and $\gamma_n \to \chi(z)\,\mu(L(z))$ as n goes to infinity, the factors of the infinite product in both (132) and (133) converge to $G(z)/z$.

2.B Appendix: The upper bound $\overline{c}_k(z)$

In this appendix, we elaborate on the pgf $\overline{c}_k(z) = \Delta_k(z, z, \ldots, z, z)$ of $\overline{c}_k$, an upper bound for the total delay experienced by a message of length $L = k$. From definition (112), we have for $k \geqslant 2$ that

$$\Delta_k(\underbrace{z, z, \ldots, z}_{k-1}, z) \;=\; \mathrm{E}\big[z^{a(J+1)} \cdot z^{a(J+2)} \cdot \ldots \cdot z^{a(J+k-1)} \cdot z^{u(J)}\big]\,. \tag{134}$$

The first $k-1$ factors in (134) can be expanded using $a(J+n) = \sum_{i=1}^{+\infty} m_i(J+n)$, $1 \leqslant n \leqslant k-1$. Then, based on system equations (97), in (134) we can relate the system state variables $m_i(J+k-1)$ $(i>1)$ in slot $J+k-1$ to variables in the previous slot $J+k-2$. In the next step, all variables $m_i(J+k-2)$ $(i>1)$ are written in terms of the variables $m_i(J+k-3)$ and so on until the only variables left in the expression are the number of new messages in the slots $J+1$ up to $J+k-1$ and the system state variables $m_i(J)$ $(i \geqslant 1)$ in slot J together with $u(J)$. The result of these calculations is

$$\begin{aligned}
&\Delta_k(z,z,\ldots,z,z) \\
&\quad= \mathrm{E}\Big[z^{m_1(J+k-1)}(zh_{11})^{m_1(J+k-2)}(zh_{12})^{m_1(J+k-3)} \cdot \ldots \cdot (zh_{1,k-2})^{m_1(J+1)} \\
&\qquad\quad \cdot z^{k-1}(h_{1,k-1})^{m_1(J)-1} \cdot (h_{2,k-1})^{m_2(J)} \cdot (h_{3,k-1})^{m_3(J)} \cdot \ldots \cdot z^{u(J)}\Big]\,,
\end{aligned} \tag{135}$$

where we have defined the functions h_{ij}, $i, j > 0$, as

$$h_{ij} = h_{ij}(z) \triangleq D_i(zD_{i+1}(\ldots zD_{i+j-1}(z)\ldots))\,, \tag{136}$$

where $h_{ij} = 1$ if either of the indices is zero, by convention. According to (44), these are known polynomials in z of degree j. Note that for $i = 1$, the functions $h_{ij}(z)$ are similar to the functions $\tilde{g}_j(z)$ defined in (72), i.e.

$$z\, h_{1j}(z) \;=\; \tilde{g}_j(z)\,, \qquad j \geqslant 1\,.$$

Next, we can use (95) to relate the number of new messages $m_1(J+n)$ generated in the slots $J+1, \ldots, J+k-1$ to the environment state $r(J+n)$ in those slots. Then, in a similar way as we did above for the variables $m_i(J+n)$, $i>1$ in the slots $J+1$ up to $J+k-1$, we can now also relate all variables $r(J+n)$ to $r(J)$ by using system equation (92), starting in slot $J+k-1$ down to slot J. After $k-1$ times applying (92), the function Δ_k now becomes:

$$\begin{aligned}
&\Delta_k(z,z,\ldots,z,z) \\
&\quad= M_0(z)\, M_0(zh_{11})\, M_0(zh_{12}) \cdot \ldots \cdot M_0(zh_{1,k-2}) \\
&\qquad \cdot r_0(\tilde{\gamma}_0)\, r_0(\tilde{\gamma}_1)\, r_0(\tilde{\gamma}_2) \cdot \ldots \cdot r_0(\tilde{\gamma}_{k-2})\, \frac{z^{k-1}}{h_{1,k-1}} \\
&\qquad \cdot \mathrm{E}\Big[\Big(Q(\tilde{\gamma}_{k-2})\Big)^{r(J)} \cdot (h_{1,k-1})^{m_1(J)}(h_{2,k-1})^{m_2(J)}(h_{3,k-1})^{m_3(J)} \cdot \ldots \cdot z^{u(J)}\Big]\,,
\end{aligned} \tag{137}$$

where the functions $\tilde{\gamma}_i$ are recalled from (73). The only remaining stochastic quantities in (137) are the system state variables for slot J, of which the joint pgf is given by $\widehat{P}(z, y_1, y_2, \ldots, z)$. Let us further define

$$\Lambda_n(z) \triangleq M_0(zh_{1n}(z))\, r_0(\tilde{\gamma}_n(z)) \qquad , n = 0, 1, \ldots k-2 \,. \tag{138}$$

We then finally find with $h_{10}(z) \triangleq 1$:

$$\begin{aligned} \Delta_k(z, z, \ldots, z, z) &= \Lambda_0(z) \cdot \Lambda_1(z) \cdot \Lambda_2(z) \cdot \ldots \cdot \Lambda_{k-2}(z) \cdot \frac{z^{k-1}}{h_{1,k-1}} \\ &\quad \cdot \widehat{P}\big(\, Q(\tilde{\gamma}_{k-2})\,,\, h_{1,k-1}\,,\, h_{2,k-1}\,,\, h_{3,k-1}\,,\, \ldots\,,\, z\,\big)\,. \end{aligned} \tag{139}$$

Now, one can obtain an explicit expression for $\Delta_k(z, z, \ldots, z, z)$ by simply substituting result (133) of Appendix 2.A.

2.C Appendix: Finite-length messages

We now point out some useful simplifications to our analysis in the particular case that the length of the messages cannot exceed a certain bound $N \geqslant 1$, as may sometimes be assumed in practical situations. In fact, all it takes to incorporate this observation is to substitute $\ell_j = 0$ for all $j > N$ into the results of the above analysis and no further problems will arise. For instance, the infinite sums in expressions (104) and (108) now reduce to finite sums and are therefore much easier to evaluate.

However, if it is known from the start that the messages are bounded, the analysis no longer requires the use of an *infinite*-dimensional description of the system state. Indeed, if no message can contain more than N packets, it is seen that $m_{n,k}$ must be 0 for all $n > N$. Therefore, the set of system state variables for slot k, as defined at the end of Sec. 2.2, reduces to $r_k, m_{1,k}, \ldots, m_{N,k}, u_{k+1}$, so no more than $N+2$ random variables are required. As a consequence, the pgf P (and also $\hat{P}$) of the system state has only $N+2$ arguments and the main functional equation (49) becomes

$$\begin{aligned} P(x, &y_1, y_2, \ldots, y_N, z) \;=\; \mathrm{E}[x^r \cdot y_1^{m_1} \cdot y_2^{m_2} \cdot \ldots \cdot y_N^{m_N} \cdot z^u] \\ &= \frac{1}{z}\, M_0(y_1 z)\, r_0\big(x\, \mu(y_1 z)\big) \\ &\quad \cdot \Big[P\left(Q\big(x\, \mu(y_1 z)\big), D_1(y_2 z), D_2(y_3 z), \ldots, D_{N-1}(y_N z), 1, z\right) + p_0\,(z-1)\Big], \end{aligned}$$

since, from (41) and (61), $D_n(z) = 1$ and $\eta_n(z) = 1$ for $n \geqslant N$. All further calculations change accordingly. Also, the iterative procedure of Appendix 2.A to obtain an explicit expression for P and $\hat{P}$ still holds. Contrary to what one might expect, an infinite number of iterations are still required, even for bounded message lengths. For $n \geqslant N$, the functions g_n in (128) become equal to $L(z)$ and no longer vary, but the functions γ_n keep changing according to (131). Thus, we still need infinite products in the results (132) and (133).

Concerning $\mathrm{E}[c]$ and $\mathrm{E}[h]$, equations (85) and (105) reveal that it might be a good idea to concentrate mainly on the calculation of those $\mathrm{E}[c_k]$ and $\mathrm{E}[h_k]$ for which ℓ_k is large (separate expressions for these conditional means can easily be derived from our discussion in Secs. 2.5.1 and 2.5.2). Anyhow, for bounded message lengths, no more than N of these terms have to be considered. Likewise, when evaluating the bounds $\overline{c}$ and $\underline{c}$ to assess the tail distribution of the message delay, only those residues $\theta^{\overline{c}_k}$ have to be determined for which $k \leqslant N$. Note that the algorithm (115)–(120) can only begin to converge beyond N iterations.

Chapter 3

The Stop-and-Wait ARQ protocol over a channel with correlated errors

In this chapter we study the performance of the Stop-and-Wait ARQ protocol for the reliable transmission of packets over an inherently *un*reliable channel. In order to do so, we propose a discrete-time model of the transmitter queue assuming the transmission errors in the channel occur in a correlated manner. Specifically, we use the so-called *Gilbert-Elliott* channel model in which the probability that a packet error occurs is modulated by a two-state Markov chain. The analysis we propose is based on the identification of a multidimensional description of the system's state, which determines the system in the Markovian sense as explained in chapter 1. In this way, we obtain results for the throughput of the protocol and the equilibrium distribution of both the number of packets present in the system and their delay.

3.1 ARQ protocols

3.1.1 Data transmission over an unreliable channel

Let us consider the situation in which a user wishes to convey information in the form of packets from point A (the *transmitter*) to point B (the *receiver*). The packets are temporarily stored at the transmitter side if necessary, and then are sent into the medium between A and B (the *channel*), one after the other. Once the packets reach the receiver, they are delivered to the end user in the correct order, i.e. in the order they were originally offered to the transmitter by the user in point A. No matter how well the channel is implemented, it can never be made a 100% reliable and it is therefore inevitable that every now and then, a packet gets damaged or is lost entirely. In many practical situations, such *transmission errors* occur more as a rule rather than exception.

In order to send the packets from A to B in a reliable way, notwithstanding the intrinsic *un*reliability of the channel, one can use a method based on *error detection* and *retransmission*, or ARQ (Automatic Repeat reQuest). With this technique, if the receiver detects an error in the packet it receives, it will request the transmitter to transmit a *new* copy of the same packet (retransmission). This implies that a copy of the packet has to remain available at the transmitter side as long as the receiver can request its retransmission. Note that there are two prerequisites in order to be able to apply ARQ:

- First, there must be some way to *check* the integrity of a packet when it reaches the receiver. For this purpose, the packets are being protected with an *error-detecting code* C_0 before they are sent into the channel. Usually, the code C_0 calculates a number of parity bits from the data in a packet from the user and appends them at the end of that packet. Then, the packet with both user data and the redundant parity bits is transmitted over the channel. At the receiver side, the packet is decoded according to C_0, which means that the parity bits are used to decide whether an error occurred in the packet or not. Note that in practice, there is still a small chance that the packet was corrupted during its transmission over the channel, but that C_0 is not able to detect this error[1]. However, in principle, the probability of an undetected error can be kept arbitrarily small by using a stronger code, which often means using more redundant bits.
- Secondly, the channel must be *bidirectional*, i.e. not only should communication be possible from the transmitter to the receiver but also in the opposite direction, via the *feedback channel*. This is necessary for the receiver to be able to send an acknowledgement message back to the transmitter for each packet it receives. When the concerning packet was received erroneously (i.e. when C_0 decides as such) the transmitter is notified of this error with a *negative acknowledgement* (NACK) and transmits the packet again. On the other hand, if no error was detected in the received packet, the receiver sends back a *positive acknowledgement* (ACK). The transmitter then knows that the packet was received correctly and no longer needs to retain a copy of that packet.

So in general, ARQ works as follows. The service requested by a user in point A is to convey a stream of packets to an end user in point B, in such a way that the packets are delivered correctly and in their original order. To this end, every packet is assigned a *sequence number* at the transmitter side, in increasing order of arrival to the ARQ transmitter. Since the packets must be sent over the channel one by one, the offered packets are stored temporarily in a *transmitter queue* where they await their transmission. Suppose that the packet with sequence number i is chosen, then its data is coded according to C_0 and the packet is transmitted over the channel for the first time. However, packet i remains stored in the transmitter queue because its retransmission may be required in the future. Packet i is now what we call an *outstanding* packet, i.e. a packet that is transmitted, but for which no feedback message (ACK/NACK) has yet returned. Once packet i reaches the receiver, it is decoded and the decision is made whether an error occurred during its transmission or not. The transmitter is

[1]Usually, the code C_0 is good at detecting one or at most a few wrong bits in the packet, which are the types of error pattern that are most likely to occur. However, in the unlikely event that *many* bits in the packet are switched, the probability increases that C_0 does not recognise this.

notified of this decision with an ACK/NACK message over the feedback channel. If the transmitter gets back an ACK for packet i, then this packet can be removed from the queue. However, if a NACK is sent back, the same packet i is retransmitted and kept in the queue for a possible *next* retransmission. The transmitter keeps performing retransmissions of packet i this way, until the receiver finally has a correct copy of the packet. At the receiver side, the errorless packets are delivered to the end user. However, for some types of ARQ it often occurs that a correct copy of a packet with higher sequence number is available at the receiver sooner than another packet with lower sequence number. This is e.g. the case when repeated transmissions were required for the latter packet. Since the end user usually wants the packets to be delivered in their original order, it is therefore necessary to provide a *resequencing buffer* at the receiver side. Indeed, as long as packet i is not received correctly, all other correctly received packets with sequence number higher than i must *wait* before being delivered to the end user.

For simplicity, we assume that no errors occur over the feedback channel, i.e. that the ACK/NACK messages are never corrupted. In most cases, this assumption is justified because the feedback messages are small[2] and can therefore be protected very well without significantly using up much of the channel capacity. In case the feedback messages can be corrupted or lost anyway, it is customary to use a time-out mechanism. This means that a packet is retransmitted if no valid ACK/NACK message is returned to the transmitter within a certain time period after the last (re)transmission of that packet.

In terms of the OSI reference model for layered network architectures (see chapter 1) the use of ARQ as an error control technique is usually situated at the link layer (level 2). In this context, the 'user' in point A is in fact the network layer protocol (level 3) that issues a service request to the link layer. The requested service is to deliver the packets without error and in their original order to the 'end user', which is the network layer protocol in point B. To provide this service, the link layer can then use either ARQ or another error control technique (see Sec. 3.C.2). In any case, the link layer will have to make use of the services provided by the physical layer (level 1) that takes care of the actual transmission of the information. In our terminology, the physical layer is characterised by the 'channel' and the possible services it provides are the transmission of data packets in the one direction and the returning of feedback messages in the other.

3.1.2 Three classical ARQ protocols

The specific set of deterministic rules that describe how the transmissions, retransmissions and acknowledgements are organised is called the ARQ *protocol* or *scheme*. Traditionally, there are three basic ARQ protocols, each with their own specific advantages and drawbacks. In aid of the following discussion, let us define the *feedback delay* s of the communication channel as the minimal time *after* a packet has been transmitted for the ACK/NACK message of this packet to return to the transmitter. We further assume that the feedback delay is fixed, i.e. the same for all packets although

[2] A feedback message contains only the binary decision ACK/NACK and the sequence number of the packet.

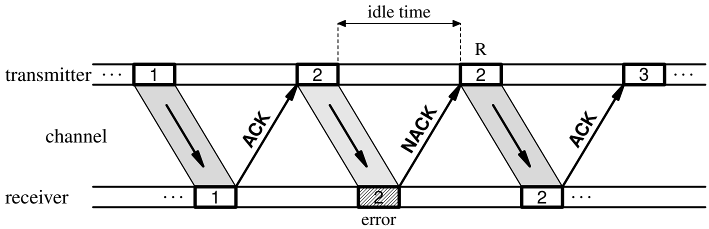


Figure 3.1: Working principle of the Stop-and-Wait ARQ protocol. An error occurs on the first transmission of packet 2, after which the packet is retransmitted (R).

for applications like e.g. satellite communication, a variable feedback delay is not unthinkable.

Stop-and-Wait ARQ

The principle of the Stop-and-Wait ARQ protocol, SW-ARQ or SW for short, is illustrated in Fig. 3.1. After the transmission or retransmission of the packet with sequence number i, the transmitter becomes idle and simply *waits* until the acknowledgement message for this packet has returned. In case of a NACK, the packet i is retransmitted while in case of a positive acknowledgement (ACK) packet i can safely be removed from the transmitter queue and the next packet with sequence number $i+1$ is transmitted for the first time. The SW-ARQ transmitter is characterised by the fact that after the transmission of each packet, the channel is left unused during the whole feedback delay s, which is inefficient. On the other hand, the SW-ARQ scheme is very simple and therefore also easy to implement. Note for instance that there is never more than one outstanding packet at the transmitter side. Also, correct copies of the packets always reach the receiver in the right order, so they can be delivered to the end user directly and there is no need for a resequencing buffer.

Go-Back-N ARQ

The Go-Back-N ARQ protocol (GBN-ARQ or GBN) uses the channel in a more efficient way than SW-ARQ because instead of allowing only one outstanding packet, it uses a so-called 'window' of maximally N unacknowledged packets. As illustrated in Fig. 3.2 for $N = 3$, packets are transmitted continuously without waiting for acknowledgements, as long as there are no more than N outstanding packets. If this maximum is reached, an idle period follows during which the transmitter waits for the first returning acknowledgement. If for a certain packet an ACK is returned, then this packet is removed from the transmitter queue *and* from the window of outstanding packets. In this window, there are now less than N packets so a new packet can be transmitted immediately.

Suppose however that on reception of packet i, an error is detected in this packet at the receiver side. Then, the receiver will send back a NACK to the transmitter and then *ignore* all further incoming packets with sequence number higher than i, as long as no correct copy of packet i has been received. The transmitter on his part, if a NACK

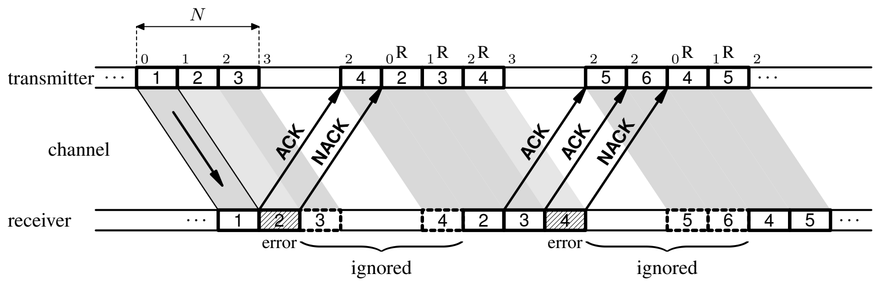


Figure 3.2: Operation of Go-Back-N ARQ with $N = 3$. The small digits indicate the number of outstanding packets in the window.

is returned for packet i, will retransmit this packet and subsequently also all *other* packets in the window, i.e. the packets with sequence number $i+1$, $i+2$ to maximally $i+N-1$. As such, in case of a negative acknowledgement for a packet, the protocol 'goes back' in time to the moment of this packet's last transmission and repeats everything it did since then. For $N = 1$, GBN is exactly the same as SW, but for $N \geqslant 1$ GBN has a higher efficiency. The larger the maximal number of outstanding packets N, the smaller the idle periods and the higher the used channel capacity. However, as is clear from Fig. 3.2, the efficiency will not increase further once N becomes larger than the number of packets that can be transmitted during one feedback delay. It is therefore useless to make the size N of the window larger than this specific number of packets, because the window would never be fully occupied then anyway. Sometimes, GBN with N sufficiently large is also called Go-Back-∞ or GB(∞)[3]. Finally, note that GBN is designed specifically to conserve the packet ordering, so just like SW, no resequencing at the receiver is needed.

Selective-Repeat ARQ

The Selective-Repeat ARQ protocol (SR-ARQ or SR) further increases the efficiency by dropping the upper bound for the number of outstanding packets on the one hand, and by no longer requiring that the protocol automatically preserves the correct packet order on the other. As shown in Fig. 3.3, the transmitter continuously transmits new packets into the channel. This is only interrupted if a NACK is returned for a previously transmitted packet. This packet is then retransmitted immediately, after which the transmitter simply continues with the transmission of new packets. So in comparison to GB(∞), if the transmitter gets back a NACK for packet i, it only retransmits this one packet and not all of the other outstanding packets as well. Also, packets are never ignored. So once they reach the receiver without error, they never have to be retransmitted. As a consequence, a correct copy of packets with sequence number higher than i may be available at the receiver *before* packet i. So contrary to SW and

[3]Often, the term Go-Back-N ARQ is used for what is in fact GB(∞). In other words, it is tacitly assumed that the window is always large enough so the transmitter never has to turn idle because of a full window.

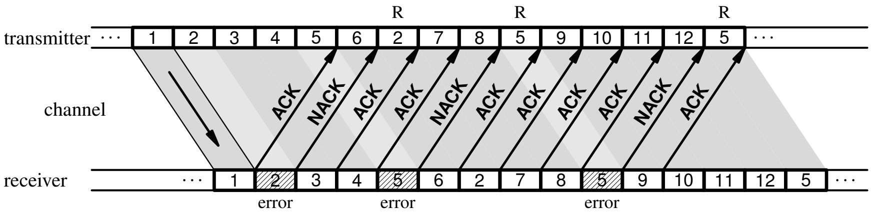


Figure 3.3: The Selective-Repeat protocol with $s = 4$. Resequencing is clearly required, since e.g. packets 6 to 12 are received correctly prior to packet 5.

GBN, the SR-ARQ scheme does not preserve the packet order and thus needs a resequencing buffer at the receiver side. In Fig. 3.3 for example, two retransmissions are required for packet 5 and the receiver obtains a correct copy of this packet only *after* it has received packets 6 to 12. It is seen that SR-ARQ allows the highest efficiency of the three classical protocols, but is also the most difficult one to implement.

3.1.3 Performance and modelling of ARQ protocols

It is clear from the discussion above that each ARQ scheme has its own advantages and drawbacks. The question is then how we should compare different protocols with each other in order to decide which one is best in a given situation. Obviously, an important criterium is the relative *performance* difference of the various schemes in this situation. To be able to evaluate the performance of an ARQ protocol in a sensible and comparative way, it is necessary to construct a *model* that makes unambiguous quantitative assumptions regarding the operating environment, as discussed in Sec. 1.3.1.

In our model, we consider the ARQ protocol in terms of the packets that are transmitted and retransmitted, which in a network, is mostly organised at the level of the link layer. As in the other chapters, we use a model in discrete time and assume that a *slot* is the fixed amount of time required to send one packet into the channel. The elements that influence the operation of the protocol and hence, need to be included in the system model, can be divided in three levels. From the lowest level to the highest level, these are:

► *The channel* (the *physical* layer)

As the primary goal of an ARQ protocol is to send packets over a communication channel, its operation obviously depends on the characteristics of that channel. We already mentioned a first parameter: the *feedback delay* s, which indicates the time required for a packet to travel from the transmitter to the receiver, to be checked for errors and for an acknowledgement to travel back. In our model, we assume that the feedback delay of every packet is the same fixed number of slots.

Secondly, we must also make specific assumptions with regard to the *errors* that occur in the channel. Note that the frequency, the typical patterns and the statistical properties of transmission errors are effectively related to the nature of the channel and the specific circumstances in which it is used. The physical causes that lead to an erroneous reception of a packet may for instance be totally different in case of a *wired* connection between A and B than in the case of a *wireless* channel medium[4]. Also, within these two classes, the error process heavily depends on the used technology, the quality of the wiring or the transmitting power of an antenna, the used modulation and coding techniques, as well as external factors such as electromagnetic interference, noise etc. Obviously, it is not possible for us to account for all these physical elements and their specific interactions in a deterministic way. Instead, we choose to *assume* that these physical causes can be captured in a stochastic *channel model* that is part of the overall ARQ system model. Specifically, in each of the subsequent slots, the channel model gives the exact *probability* that a packet will be corrupted.

► *The ARQ scheme*
The ARQ *protocol* on the other hand, is the purely *deterministic* part of the system model. It contains the specific rules that are used with regard to waiting and outstanding packets, what to do when an ACK or a NACK is returned, which packet is to be transmitted or retransmitted and at what time. For instance, we know that the packets offered by the user are temporarily stored in a transmitter queue, that this queue has a FIFO discipline, that a window has to be maintained with outstanding packets, as well as a resequencing buffer at the receiver side. So from a systemic point of view, the ARQ protocol is a dynamic *data structure* (transmitter queue and resequencing queue) together with a set of event-based rules that tell exactly how to react to every possible situation. In case of SW, GBN and SR, these rules have been outlined in the previous section.

► *The users*
We have the same problem with the level *above* the system as with the channel characteristics below. It is not possible, nor desirable, to describe deterministically how many packets the users will generate in each of the subsequent slots, so we must use a stochastic model here as well. Obviously, this relates to the packet *arrival process* of the queueing model in the sense of chapter 1.

Constructing a model always implies making simplifying assumptions; so not all features that influence the ARQ operation are included in the model. The goal of the model in this chapter is to focus mainly on the *retransmission* aspect of ARQ, rather than e.g. the *coding* aspect. As such, we assume that the error-detecting code C_0 which protects the packets is perfect, so when a transmission error occurs, the receiver will always be able to detect this[5]. Also, the assumption that the return channel is without errors has been mentioned before.

The structure of a typical ARQ system and the route of a packet P being handled by this system is depicted in Fig. 3.4. There is a transmitter queue at the transmitter side, possibly also a resequencing buffer at the receiver side and the channel in between. If P arrives at the transmitter side, it is stored in the queue, waiting to be transmitted for the first time. At that moment, P becomes an 'outstanding' packet and is retransmitted until it is received correctly. The transmitter is notified of this and the copy of P

[4] We will discuss this further in Sec. 3.C

[5] This assumption is not really essential for the evaluation of ARQ at the level of the link layer. In practice, undetected erroneous packets are often mitigated by an upper layer protocol (e.g. TCP at the transport layer) or are simply considered to be harmless (in the case of e.g. real-time applications).

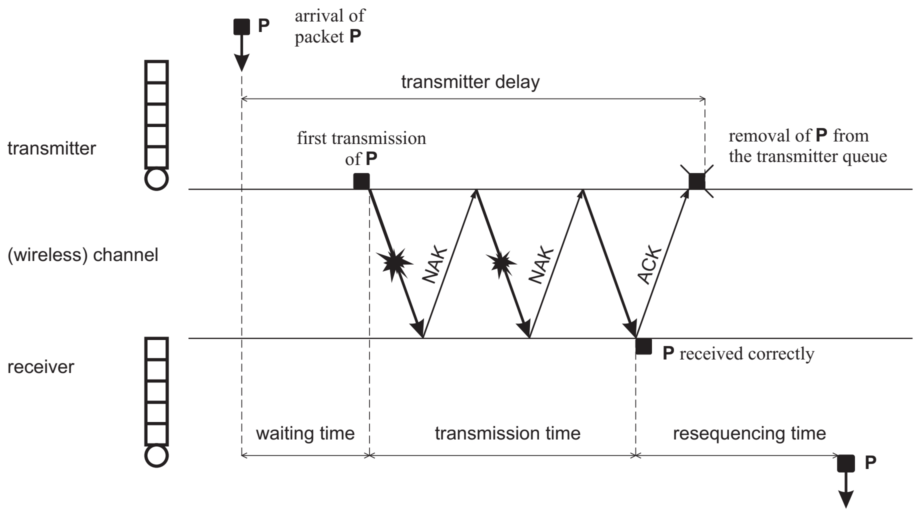


Figure 3.4: A general ARQ system with the relevant delay times for a packet P.

leaves its queue. At the receiver, P must wait again until all other packets with a serial number lower than that of P are received correctly as well, and can then finally be delivered to the end user. Some of the results we want to obtain from such a model are the following.

► *Throughput*
The maximal throughput of the system can be defined as the average number of packets per slot that are received correctly in case there are always packets available in the transmitter queue. The throughput is a quantitative measure for what we have earlier called the *efficiency*.

► *The content of the transmitter queue*
The number of packets present in the queue, i.e. the number of packets that are either waiting for their first transmission or that are outstanding.

► *Transmitter delay of the packets*
Directly related to the number of packets present in the queue is also the delay that is experienced by these packets. Specifically, the measure of interest is the *transmitter delay*, i.e. the total amount of time that (a copy of) the packet spends in the transmitter queue.

In the following analysis, (the distribution of) each of these performance measures is calculated exactly for the SW-ARQ protocol under equilibrium conditions. Note that we are not concerned with the resequencing delay, as no resequencing is required for this protocol.

3.2 Stop-and-Wait performance over a Gilbert-Elliott channel

As we have seen, for SW-ARQ the transmitter is inactive during the whole time a packet is travelling through the channel, is being processed by the receiver and while the feedback message is travelling back. We have referred to this time period as the *feedback delay* s and it is clear that for long such delays, quite some time is wasted simply waiting for acknowledgements, resulting in a low throughput. However, SW-ARQ is simple to implement and ensures that packets are received in the same order as they arrived to the transmitter such that no resequencing is needed.

The model we propose distinguishes itself from previous studies in that we allow the errors occurring in the channel to be *correlated in time*. Instead of assuming that the probability of an erroneous packet is static in time, we propose that this probability depends on what *state* the channel is in when the packet is transmitted. Specifically, the channel alternates between two states which could be termed the GOOD state and the BAD state, both of which reflect different conditions with regard to the error probability. The channel state process is modelled as a two-state Markov Chain with a fixed error probability in either state, resulting in what is also known as the Gilbert-Elliott model [76, 97]. This complication is inspired by the observation that real-life communication channels rarely have the same error sensitivity during their whole time of operation. Factors such as electromagnetic interference, availability of intermediate network nodes and links, presence of data traffic with higher priority and so on, may all influence the behaviour of the channel and are mostly time-varying in nature. This holds especially when the *wireless* medium is considered where the conditions may change on an even more diverse set of timescales than in a wired medium due to user mobility, noise, reflection, scattering, shadowing, or any other physical effect that causes the radio signal to change in both amplitude and phase. Such inherent signal changes over time and space impairing the data transmission are known as *channel fading*, see Sec. 3.C.1 for further details.

The Gilbert-Elliott channel model with only two states certainly has its merits when used for the analysis of ARQ protocols, see e.g. [48, 224] as well as the examples in [189]. As demonstrated in this chapter, some of the *qualitative* effects on the queueing performance of the protocol that are caused by the correlation in the error process are manifested in their purest form precisely when using only two states. The distinction made between symmetric and compensated stability in Sec. 3.7.2 is an example. We also note that in principle, our analytic technique can equally well be used in the case of a model with N states, although explicit results for the queue content and delay distribution can in general only be provided for $N = 2$.

The performance of SW-ARQ has been studied before, both in terms of throughput and queueing behaviour, but almost always in case of static error probabilities. Several modifications have been proposed to enhance the performance of SW-ARQ, such as sending multiple copies of a packet during the time the transmitter is waiting for feedback [48, 79, 146, 151] or combining ARQ with improved error correction techniques, known as hybrid ARQ [150]. In [48], the influence of decoding with memory has been studied as well. Another improvement is the SW-ARQ protocol suggested in [192], where each packet is divided in n parts and only those parts that were received in

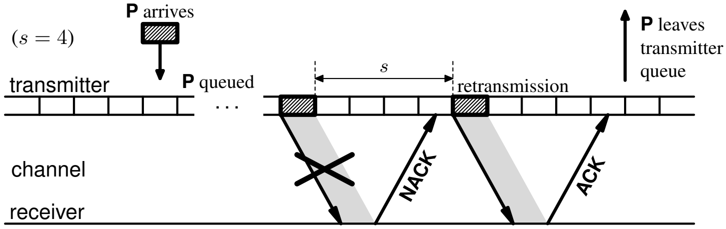


Figure 3.5: Operation of the transmitter queue under SW-ARQ, meaning of the feedback delay s.

error are retransmitted. [170] studies the throughput and packet delay distribution of all three major ARQ protocols over a TDMA channel with static errors, although for Selective-Repeat ARQ (SR-ARQ), only bounds for the mean packet delay are given. In [81], the transmitter queue behaviour of SW-ARQ is analysed in a continuous-time setting. Towsley [187, 189] has worked on the SW-ARQ model with correlated errors as well, but did not have results for the distribution of the packet delay as we do in Sec. 3.6. Also, our analysis is quite different on several accounts and we provide more explicit results in case of a two-state error channel.

3.3 Model description

We model the transmitter of a system operating under the Stop-and-Wait (SW) ARQ protocol as a discrete-time queue. As usual, we assume that time is divided in fixed-length intervals called *slots*, whereby one slot is the time required to transmit one packet from the queue into the channel. In our analysis, the length of a certain time period is always expressed as an (integer) number of slots. Let us assume the feedback delay is a fixed number of slots too, denoted by s. The operation of SW-ARQ is illustrated in Fig. 3.5. A packet P arrives in the system and is queued for some time until all preceding packets are transmitted correctly and the ACK of the previous packet is received. Then P is transmitted for the first time. If the transmission is erroneous, a NACK is returned and P is retransmitted $s{+}1$ slots after its previous transmission. The packet is retransmitted until finally, an ACK is returned. Then the transmitter knows P was transmitted correctly and there is no need to keep it in the queue any longer. One may wonder if the feedback messages can be corrupted too, since they use the same error-prone channel as the data packets. However, these messages are usually small and easier to protect than the data packets. Hence, our model does not account for erroneous ACK/NACK messages, which is in fact a non-restrictive assumption [168].

Packets of information enter the system according to a *general independent arrival process*, i.e. the numbers of arrivals during consecutive slots form a sequence of independent and identically distributed random variables with common mass function $a(n) = \text{Prob}[a = n]$ $(n \geqslant 0)$ and probability generating function $A(z) = \sum_{n=0}^{\infty} a(n) z^n$. Let a_k be the number of arriving packets during slot k, then we denote its mean value by $\alpha \triangleq \mathrm{E}[a_k] = A'(1)$. Furthermore, we assume that the arrivals a_k are not stored in the queue until the end of slot k. This way, an arriving packet can only be served (i.e.

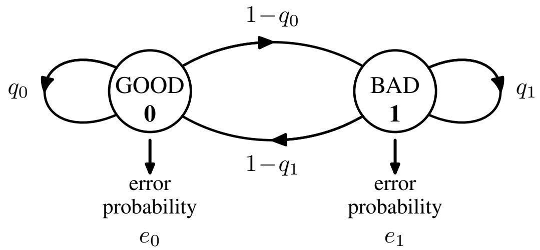


Figure 3.6: Markovian error model for the transmission channel.

transmitted) for the first time during the *next* slot ($k+1$) at the very earliest.

When a packet is transmitted, its successful receipt depends on the channel state during the slot in which the feedback message is returned to the transmitter. The transitions between those states, the 0-state (GOOD) and the 1-state (BAD), are governed by a two-state Markov Chain as depicted in Fig. 3.6: if the feedback of a packet is returned during a 0-slot, this means the packet was transmitted erroneously with probability e_0. Similarly, during a 1-slot a feedback message is a NACK with probability e_1 and ACK with probability $1-e_1$. In other words, if we adopt the notation $\bar{q} \triangleq 1-q$, then $\bar{e}_i$ is the probability of a correct transmission, $i = 0, 1$. Evidently, the designations GOOD and BAD are only meaningful if $e_0 < e_1$, although this condition is not a requirement for the analysis. At first it may seem strange that we choose not to probe the channel state during the slot in which the packet *is transmitted* but during the slot in which its feedback *is returned* to the transmitter. However, this modelling choice makes the analysis less complicated while the results regarding the equilibrium behaviour stay the *same*. Indeed, it is only a matter of definition to which *actual* slot k–s we refer to by the name 'channel state in slot k'. Since the evolution of the channel state is not influenced by the rest of the system, the results of the analysis are not affected by a fixed time shift of this definition.

First of all, as a convention for the remainder of this chapter, let the index i always be either 0 or 1. As indicated in Fig. 3.6, the probability of remaining in state i during a slot transition is given by q_i. We denote the channel state (0 or 1) in slot k by r_k and let $\omega_{i,k} \triangleq \text{Prob}[r_k = i]$. Then we have

$$\boldsymbol{\omega}_{k+1} = \boldsymbol{\omega}_k \mathbf{q} \qquad \text{with} \quad \mathbf{q} \triangleq \begin{bmatrix} q_0 & \bar{q}_0 \\ \bar{q}_1 & q_1 \end{bmatrix} , \tag{140}$$

where $\boldsymbol{\omega}_k$ is the row vector with elements $\omega_{0,k}$ and $\omega_{1,k}$ and $\mathbf{q}$ is the transition probability matrix of the channel state process. Since the channel states are Markovian, both the 0-periods and 1-periods are geometrically distributed, i.e.

$$\text{Prob}[i\text{-period of length } n] \;=\; (1-q_i)q_i^{n-1} \qquad , \, n \geqslant 1 \, .$$

Also from Fig. 3.6, we have for the conditional pgfs of r_k:

$$\begin{aligned} r_0(z) &\triangleq \text{E}[z^{r_{k+1}} | r_k = 0] \;=\; \bar{q}_0 z + q_0 \, , \\ r_1(z) &\triangleq \text{E}[z^{r_{k+1}} | r_k = 1] \;=\; q_1 z + \bar{q}_1 \, ; \end{aligned} \tag{141}$$

where $\mathrm{E}[X|Y]$ is the expected value of X given that Y holds. It turns out that the lengths of 0- and 1-periods (i.e. the values q_0 and q_1) determine the performance of the queue in a crucial way. However, rather than using q_0 and q_1, we define the parameters

$$\sigma = \frac{1-q_0}{2-q_0-q_1} \quad \text{and} \quad K = \frac{1}{2-q_0-q_1},$$

to be understood as follows. Suppose the channel is in state 0 with probability $\bar{\sigma}$ and in state 1 with probability σ, independently from slot to slot, such that the mean sojourn times are $1/\sigma$ and $1/\bar{\sigma}$ respectively. It is clear that the overall fraction of 1-slots remains equal to σ if the mean lengths of 0- and 1-periods are both multiplied by the same factor K, i.e. if the geometric distributions are chosen such that the mean lengths are $1/(1-q_0) = K/\sigma$ and $1/(1-q_1) = K/\bar{\sigma}$ respectively. Therefore, the factor K can be seen as a measure for the *absolute* lengths of the 0- and 1-periods, while σ characterises their *relative* lengths. Indeed, σ is the relative fraction of 1-slots, since we have from (140) in equilibrium:

$$\boldsymbol{\omega} = \lim_{k\to\infty} \boldsymbol{\omega}_k = \begin{bmatrix} \bar{\sigma} & \sigma \end{bmatrix} . \tag{142}$$

Moreover, the correlation coefficient ϕ between the channel states in two consecutive slots in equilibrium is determined by the value K:

$$\phi = \lim_{k\to\infty} \frac{\mathrm{E}[r_k\, r_{k+1}] - \mathrm{E}[r_k]\mathrm{E}[r_{k+1}]}{\sqrt{\mathrm{Var}[r_k]\mathrm{Var}[r_{k+1}]}} = -1+q_0+q_1 = 1 - \frac{1}{K} .$$

Note that $\phi = 0$ and $K = 1$ for uncorrelated errors, whereas for positive correlation we have $0 < \phi < 1$ and $K > 1$. The more correlation present in the channel state process $\{r_k\}$, the higher K is and the fewer the channel changes state. For example, if $\sigma = 0.5$, typical samples of $\{r_k\}$ may be :

011000110101000110111001101001001101000110111 K small,
000000001111111111100000001111111111000000 K large.

3.4 Distribution of system state and queue content

Our study of the transmitter queue described above is done by using the DSVT method explained in 1.3.4. This means we model the system as a (multi-dimensional) Markov Chain and calculate its equilibrium distribution, assuming such equilibrium exists. Hence, we first need a description of the *system state* in the Markovian sense, i.e. a set of random variables such that their distribution in slot $k+1$ depends *only* on the values of the corresponding variables in slot k (and not on those in previous slots). The set must also be sufficiently large to solve the problem at hand. In other words, what is the information we need to keep track of if we can only look back one slot?

As the system state variables, we choose the set illustrated in Fig. 3.7, constructed as follows. Let u_k be the queue content at the beginning of slot k, which obviously needs to be included since $\{u_k\}$ is the process we are interested in. Next, we also need to know how far a packet has progressed through the channel during slot k and when

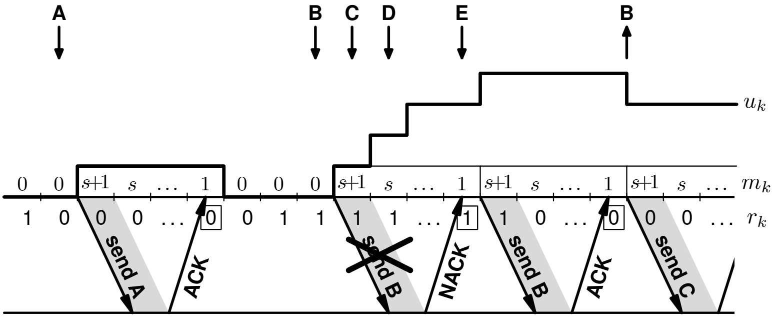


Figure 3.7: Evolution of the system state given by the channel state r_k, the residual roundtrip time m_k and the queue content u_k.

we can expect its feedback message. For this purpose we define the supplementary variable m_k, in a similar way as was done in [44]. The *residual roundtrip time* m_k indicates the remaining number of slots at the beginning of slot k, needed to complete the roundtrip of the most recently transmitted packet if $u_k \geqslant 1$, and $m_k = 0$ if and only if $u_k = 0$. So $m_k = s+1$ when a packet is transmitted and then counts one down in each of the following slots. After s slots, when $m_k = 1$, we know that the feedback message for this packet is being returned and the packet will either leave the queue at the end of the slot (ACK) or be retransmitted in the next slot (NACK). Finally, as before, let the random variable r_k be the channel state during slot k. The channel state comes into play during slots with $m_k = 1$, where it determines the probability that either an ACK or a NACK is returned, or equivalently, that the packet departs from the queue or is to be retransmitted. Let d_k be equal to 1 if a packet departs at the end of slot k and equal to 0 otherwise. Then if $m_k = 1$, the probability of an error, and therefore of '$d_k = 0$', is e_i if $r_k = i$. Hence, the pgf of d_k is given by $d_i(z) = \bar{e}_i z + e_i$ if $r_k = i$. We also define

$$\bar{d}_i(z) \triangleq e_i z + \bar{e}_i \,, \tag{143}$$

which is the pgf of $\bar{d}_k = 1 - d_k$.

3.4.1 System equations

One verifies that the triple $\{r_k, m_k, u_k\}$ is an adequate Markovian description of the system state at the beginning of slot k. The transitions from slot k to slot $k+1$ in this (three-dimensional) Markov Chain are described by the following set of system equations. We specify of what possible events in slot k the events '$m_{k+1} = h$', $h = 0, 1, \ldots, s, s+1$ can be the result. Note that in all cases, the transitions for r_k are in correspondence with (141).

► $m_{k+1} = 0$ (and therefore $u_{k+1} = 0$):
An empty system in slot $k+1$ can be the result of two distinct events in slot k. Either the queue was empty in slot k as well, or the queue contained exactly one packet which departed. Obviously, no new packets should arrive in either case:

$$\text{`}m_{k+1} = 0\text{'} \;=\; \text{`}m_k = 0, a_k = 0\text{'} \vee \text{`}m_k = 1, u_k = 1, d_k = 1, a_k = 0\text{'} \,. \tag{144}$$

- $0 < m_{k+1} < s+1$:
  In this case, we know that in slot k the system is not empty and that there can be no departures, since no feedback message is expected. Hence:

$$m_{k+1} = m_k - 1\,, \tag{145}$$

$$u_{k+1} = u_k + a_k\,. \tag{146}$$

- $m_{k+1} = s+1$:
  In this case, a new roundtrip period starts with the transmission of a packet in slot $k+1$. Therefore, in slot k, the queue was either empty but with the arrival of new packets, or it was at the last slot of a roundtrip period ($m_k = 1$). However, in the latter case we have to exclude the possibility mentioned in (144) that the roundtrip ends with a departure and there is no packet available to start a new one in slot $k+1$:

$$\begin{aligned} \text{`}m_{k+1}\!=\!s\!+\!1\text{'} &= \text{`}m_k\!=\!0, a_k > 0\text{'} \\ &\quad \vee \text{`}m_k\!=\!1, \neg(u_k\!=\!1, d_k\!=\!1, a_k\!=\!0)\text{'}\,. \end{aligned} \tag{147}$$

  In either case we have:

$$u_{k+1} = u_k - d_k + a_k\,. \tag{148}$$

3.4.2 Equilibrium distribution of the system state

We use the above relations to obtain separate expressions for the distribution of the system state during slots where the queue is *idle* ($m_k\!=\!0$) and during slots where the queue is *busy* ($m_k > 0$). First, let us define $p_{i,k}$ as the probability that the queue is empty and that the channel is in state i in slot k,

$$p_{i,k} \triangleq \text{Prob}[m_k\!=\!0, r_k\!=\!i]\,. \tag{149}$$

Secondly, let $yzH_{i,k}(y,z)$ be the joint partial pgf of the residual roundtrip time and the queue content in slot k for a busy queue and channel state i in that slot:

$$H_{i,k}(y,z) \triangleq \text{E}[y^{m_k-1} z^{u_k-1}\{m_k > 0, r_k\!=\!i\}]\,, \tag{150}$$

where we use the notation $\text{E}[X\{Y\}] = \text{E}[X|Y]\,\text{Prob}[Y]$. Additionally, we define $zR_{i,k}(z)$ as the partial pgf of the queue content in slot k for the case it is the last slot of a roundtrip period ($m_k\!=\!1$) and the channel state is i:

$$R_{i,k}(z) \triangleq \text{E}[z^{u_k-1}\{m_k\!=\!1, r_k\!=\!i\}] = H_{i,k}(0,z)\,. \tag{151}$$

Note that $H_{i,k}(y,z)$ and $R_{i,k}(z)$ are *partial* pgf's such that

$$R_{i,k}(1) = \text{Prob}[m_k\!=\!1, r_k\!=\!i] < H_{i,k}(1,1) = \text{Prob}[m_k > 0, r_k\!=\!i] < 1\,.$$

We proceed by using the system equations (141) and (144)–(148) to devise a relation between $p_{i,k+1}$ and $p_{i,k}$ on the one hand, and $H_{i,k+1}(y,z)$ and $H_{i,k}(y,z)$ on the other. First, from (149), and subsequently using (144), the uncorrelated nature of the arrival process, (141) and (151), we find

$$\begin{aligned} p_{0,k+1} + xp_{1,k+1} &= \mathrm{E}[x^{r_{k+1}}\{m_{k+1}=0\}] \\ &= A(0)\Big(\mathrm{E}[x^{r_{k+1}}\{m_k=0\}] + \mathrm{E}[x^{r_{k+1}}\{m_k=1, u_k=1, d_k=1\}]\Big) \\ &= A(0)\Big(\sum_{i=0}^{1} r_i(x)p_{i,k} + \sum_{i=0}^{1} r_i(x)\bar{e}_i R_{i,k}(0)\Big)\,. \end{aligned} \tag{152}$$

Secondly, from (150), and using the equations (145)–(148) we have

$$\begin{aligned} &H_{0,k+1}(y,z) + xH_{1,k+1}(y,z) \\ &\quad= \mathrm{E}[x^{r_{k+1}}y^{m_{k+1}-1}z^{u_{k+1}-1}\{m_{k+1}>0\}] \\ &\quad= y^{-1}\mathrm{E}[x^{r_{k+1}}y^{m_k-1}z^{u_k+a_k-1}\{1<m_k\leqslant s+1\}] \\ &\qquad+ \mathrm{E}[x^{r_{k+1}}y^{s}z^{a_k-1}\{m_k=0, a_k>0\}] \\ &\qquad+ \mathrm{E}[x^{r_{k+1}}y^{s}z^{u_k+a_k-d_k-1}\{m_k=1, \neg(u_k=1, d_k=1, a_k=0)\}]\,, \end{aligned} \tag{153}$$

which can readily be written in terms of $p_{i,k}$, $R_{i,k}(z)$ and $H_{i,k}(y,z)$ using (141) and the definitions (149)–(151).

We assume that for $k\to\infty$, the system reaches equilibrium, such that $p_{i,k}$ and the functions $R_{i,k}(z)$ and $H_{i,k}(y,z)$ converge to a limiting value which we indicate by dropping the index k. Also, let r, m, u and a denote the channel state, the remaining roundtrip time, the queue content and the number of arrivals respectively, in an arbitrary slot during equilibrium. The condition for the system to reach such an equilibrium will be discussed further in Sec. 3.5. Due to (152), the partial pgf of the channel state in equilibrium for an idle queue, is thus implicitly given by

$$p_0 + xp_1 = A(0)\sum_{i=0}^{1} r_i(x)\big(p_i + \bar{e}_i R_i(0)\big)\,. \tag{154}$$

Likewise, we find from (153) the following relation for the equilibrium distribution of the system state for a busy queue:

$$\begin{aligned} H_0(y,z) + xH_1(y,z) &= \sum_{i=0}^{1} r_i(x)\Big[\frac{1}{y}A(z)\big(H_i(y,z) - R_i(z)\big) \\ &\quad+ \frac{y^s}{z}A(z)\big(p_i + \bar{d}_i(z)R_i(z)\big) - \frac{y^s}{z}A(0)\big(p_i + \bar{e}_i R_i(0)\big)\Big]\,. \end{aligned} \tag{155}$$

The relations (154) and (155) constitute a set of (functional) equations that implicitly determine the distribution of the system state in equilibrium, i.e. the probabilities p_i and the functions $H_i(y,z)$. Let us arrange these quantities in the row vectors $\mathbf{p}$ and $\mathbf{H}(y,z)$ respectively as

$$\mathbf{p} = \begin{bmatrix} p_0 & p_1 \end{bmatrix}, \qquad \mathbf{H}(y,z) = \begin{bmatrix} H_0(y,z) & H_1(y,z) \end{bmatrix}. \tag{156}$$

Fortunately, it is possible to solve (154) and (155) for the functions $H_i(y,z)$. We observe that these equations are polynomials of degree 1 in x. Therefore, we can identify the coefficients of x on both sides of either equation. For (154), this yields the following two relations between the probabilities p_0, p_1 and $R_0(0), R_1(0)$:

$$\begin{aligned} A(0)\big(\bar{e}_0 R_0(0) + \bar{e}_1 R_1(0)\big) &= (1-A(0))\big(p_0+p_1\big)\,; \\ \phi A(0)\big(\sigma\bar{e}_0 R_0(0) - \bar{\sigma}\bar{e}_1 R_1(0)\big) &= (1-\phi A(0))\big(\sigma p_0-\bar{\sigma} p_1\big)\,. \end{aligned} \tag{157}$$

In the same way, we find these explicit expressions for $H_i(y,z)$ from (155):

$$\begin{aligned} &(y-A(z))(y-\phi A(z))H_i(y,z) \\ &\quad= \frac{A(z)}{z}\Big[y\bar{q}_{\bar{i}}\big(y^{s+1}\bar{d}_{\bar{i}}(z)-z\big)R_{\bar{i}}(z) + (yq_i-\phi A(z))\big(y^{s+1}\bar{d}_i(z)-z\big)R_i(z)\Big] \\ &\qquad+ \frac{y^{s+1}}{(1-\phi)z}\Big[\bar{q}_{\bar{i}}(\phi A(z)-y)(1-A(z))(p_0+p_1) \\ &\qquad\qquad+ (\phi A(z)-1)(y-A(z))(\bar{q}_i p_i-\bar{q}_{\bar{i}} p_{\bar{i}})\Big]\,, \end{aligned} \tag{158}$$

where we already have included the relations (157) to eliminate the probabilities $R_i(0)$. The only remaining unknowns in (158) are the probabilities p_i and the functions $R_i(z)$. An expression for the latter can be obtained using the following property of (partial) pgfs (see also [44]). The functions $H_i(y,z)$, as a result of their definition (150), must be bounded for all values of y and z with $|y|\leqslant 1$ and $|z|\leqslant 1$. In particular, because $A(z)$ is a pgf, this should be the case for $y=A(z)$ and $y=\phi A(z)$ with $|z|\leqslant 1$, since $\phi\leqslant 1$ and $|A(z)|\leqslant 1$ for all such z. Now, if we choose either $y=A(z)$ or $y=\phi A(z)$ in (158) where $|z|\leqslant 1$, the left hand side of this equation vanishes. Since $H_i(y,z)$ is bounded for these values of y, the right hand side must be equal to zero too. Applying the substitutions $y=A(z)$ and $y=\phi A(z)$ into (158) for either $i=0$ or $i=1$ yields the *same* set of two relations between $R_0(z)$ and $R_1(z)$ from which these functions can be determined as:

$$\begin{aligned} R_i(z) &= \frac{A(z)^s}{N(z)}\Big[\phi^s(\phi A(z)-1)\big(A(z)^{s+1}\bar{d}_{\bar{i}}(z)-z\big)(\bar{q}_{\bar{i}} p_{\bar{i}}-\bar{q}_i p_i) \\ &\qquad- \bar{q}_{\bar{i}}(A(z)-1)\big((\phi A(z))^{s+1}\bar{d}_{\bar{i}}(z)-z\big)(p_0+p_1)\Big]\,, \end{aligned} \tag{159}$$

where the denominator $N(z)$ follows as a known function of the system parameters:

$$N(z) = \sum_{i=0}^{1} \bar{q}_i\big((\phi A(z))^{s+1}\bar{d}_i(z) - z\big)\big(A(z)^{s+1}\bar{d}_{\bar{i}}(z) - z\big)\,. \tag{160}$$

3.4.3 Probability of an empty queue

At this point, the only remaining unknowns in our analysis are the probabilities p_0 and p_1 of having an empty queue while the channel is in state 0 or 1 respectively. To determine those, we need two additional relations. A first relation can be found from the normalisation condition $1 = p_0+p_1+H_0(1,1)+H_1(1,1)$, which after taking the limit $y\to 1$ and $z\to 1$ in (158), turns out to be equivalent to

$$\alpha = \bar{e}_0 R_0(1) + \bar{e}_1 R_1(1)\,, \tag{161}$$

where $\alpha = A'(1)$ is the mean number of arriving packets per slot. Again, by taking the limit $z \to 1$ in (159) and by using (161), we explicitly find

$$(1-\phi^{s+1})\big[(s+1)\alpha + (p_0+p_1-1)(\bar{\sigma}\bar{e}_0+\sigma\bar{e}_1)\big] \;=\; \phi^s(s+1)(e_0-e_1)(\bar{q}_0p_0+\bar{q}_1p_1)\,. \quad (162)$$

Note that multiple applications of de l'Hôpital's rule were required to obtain both (161) and (162). The normalisation condition (162) allows us to write all results with only a single unknown p_0+p_1 which indicates the equilibrium probability that the queue is empty. It is possible to determine this last unknown probability numerically. Indeed, a second relation for p_0 and p_1 can be obtained from (159) in a similar way as before. Again, we can use either $R_0(z)$ or $R_1(z)$ for the procedure, since both yield the same result. Because $R_i(z)$ is a (partial) pgf, it must be analytic and therefore bounded in the unit disc $|z|<1$. Hence, if we can find a value $z=z^*$ in the unit disc for which the denominator $N(z)$ of $R_i(z)$ vanishes, the numerator must also be equal to zero for $z=z^*$. We assert that it can be proven with Rouché's theorem that the denominator $N(z)$ has exactly one zero z^* inside the unit disc, i.e. $N(z^*)=0$, although the proof may not be straightforward. This zero can easily be calculated using a numerical root-finding algorithm, such as e.g. the Newton-Raphson scheme. The additional relation between p_0 and p_1 we were looking for is thus obtained by substituting z^* into the numerator of (159) and let it equal zero. The obtained expression then determines the probability p_0+p_1, which completes our analysis of the equilibrium distribution of the system state (r,m,u). From (158) and (159), one finds the (unconditional) joint pgf of the system state in equilibrium as

$$P(x,y,z) \;\triangleq\; \mathrm{E}[x^r y^m z^u] \;=\; p_0 + xp_1 + yzH_0(y,z) + xyzH_1(y,z)\,, \quad (163)$$

where the probabilities p_i can be eliminated as explained.

3.4.4 Distribution of the queue content

Obviously, if we know the joint distribution of the system state (r,m,u), we can also obtain the marginal distribution of the queue content u in equilibrium. Let $U(z)$ be the pgf of the queue content at the beginning of an arbitrary slot, then

$$\begin{aligned} U(z) \;&=\; P(1,1,z) \;=\; \mathrm{E}[z^u] \;=\; \sum_{i=0}^{1}\mathrm{E}[z^u\{r=i\}] \\ &= \sum_{i=0}^{1}\mathrm{E}[z^u\{r=i,m=0\}] + \mathrm{E}[z^u\{r=i,m>0\}] \\ &= \sum_{i=0}^{1} p_i + zH_i(1,z) \;=\; \frac{(1-z)A(z)}{1-A(z)}\big(\bar{e}_0R_0(z)+\bar{e}_1R_1(z)\big)\,, \end{aligned} \quad (164)$$

for which we took the limit $y\to 1$ in (158). After using the expressions (159) for $R_i(z)$ and the normalisation condition (162), we finally find

$$U(z) = \frac{(1-z)A(z)^{s+1}}{\big(1-A(z)\big)N(z)}\Big[z(1-\phi^{s+1})\big(1-\phi A(z)\big)\big(1-A(z)^{s+1}\big)$$

$$\cdot\Big(\alpha+\frac{\bar\sigma\bar e_0+\sigma\bar e_1}{s+1}(p_0+p_1-1)\Big)\;\;+(p_0+p_1)\bar\phi\big(1-A(z)\big) \tag{165}$$
$$\cdot\Big(\bar e_0\bar e_1(1-z)\phi^{s+1}A(z)^{s+1}-z(\bar\sigma\bar e_0+\sigma\bar e_1)\big(1-\phi^{s+1}A(z)^{s+1}\big)\Big)\Big]\,.$$

Various interesting measures concerning the behaviour of the queue content can be derived from this pgf. Invoking the moment generating property of pgfs on (165) yields closed-form expressions for the moments of the queue content up to any desired order, although the subsequent differentiations quickly become very involved for higher-order moments. For instance, the mean queue content $\mathrm{E}[u]=U'(1)$ follows as

$$\begin{aligned}\mathrm{E}[u]\;&=\alpha\Big(\frac{s}{2}-\frac{\phi}{\bar\phi}\Big)+\frac{1}{2(1-\phi^{s+1})[\bar\sigma\bar e_0+\sigma\bar e_1-(s+1)\alpha]}\\
&\quad\cdot\Big[2\phi^{s+1}\Big[(p_0+p_1)\bar e_0\bar e_1-(\bar e_0-(s+1)\alpha)(\bar e_1-(s+1)\alpha)\Big]\\
&\qquad+\alpha\frac{(p_0+p_1)}{1-\phi}(\bar\sigma\bar e_0+\sigma\bar e_1)\big((s+2)(\phi-\phi^{s+1})+s(\phi^{s+2}-1)\big)\\
&\qquad+(1-\phi^{s+1})(s+1)\big(A''(1)-(s+2)\alpha^2+2\alpha\big)\Big]\,.\end{aligned} \tag{166}$$

We defer the discussion about the tail distribution Prob$[u=n]$ for large n to Appendix 3.B.

3.5 Calculation of the throughput

We now focus on the calculation of the throughput of the SW-ARQ system described above. What follows is a new and more intuitive proof for the throughput expression already stated in [189]. Let us call the *service period* of a packet the period it stays in the server of the queue, i.e. from the slot in which it is transmitted for the first time, up to and including the slot in which it departs from the queue. The length of a service period is called the *service time* of that packet. Clearly, the service time is always an integer multiple of $s+1$, as it is always composed of a number of roundtrip periods: one for the initial transmission of the packet and possibly an additional number for retransmissions. We define the *throughput* η of the system as the maximum number of packets *per slot* that can be correctly delivered to the receiver. Hence, η is a measure for the maximum output rate at which the system can transmit incoming packets and should therefore be compared to α, the mean arrival rate. Indeed, in order to have a stable queue that reaches equilibrium, we must have

$$\alpha<\eta\,,\qquad\text{or}\qquad\rho\triangleq\frac{\alpha}{\eta}<1\,, \tag{167}$$

where we call ρ the *load* of the system. Obviously, the maximum output rate is only achieved if the system operates under overload conditions, i.e. if we assume there are *always* packets waiting in the queue for transmission. Thus, as soon as a packet is ACKed and leaves the queue, there is always another packet available to transmit immediately. Under these conditions, we can also define the throughput η as the inverse

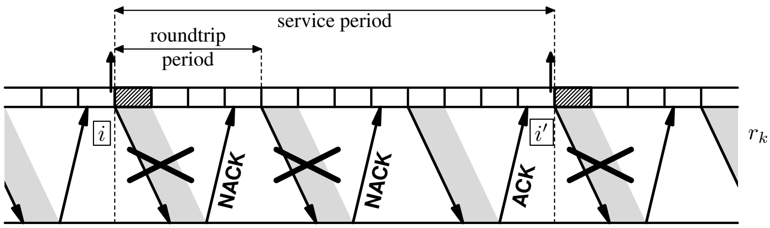


Figure 3.8: The service time of a packet of $n = 3$ roundtrip periods starting in channel state i and ending in state i'.

of the mean service time of an arbitrary packet. Therefore, we proceed by deriving the service time distribution of the packets. For this, we do not need to consider the queued packets as in the previous section, but only the server and the state of the channel.

Unfortunately, as a consequence of the correlated nature of the channel, the service times are *not independent*. Specifically, a packet's service time distribution will generally be different depending on whether its initial transmission happens during a 0-slot or during a 1-slot. Let us say that the service time of a packet *starts* in channel state i if the channel is in state i in the slot *before* the initial transmission of the packet, as indicated in Fig. 3.8. Conversely, the service *ends* in channel state i' if the channel state is i' in the slot where the packet leaves the queue. Now, let us define $\gamma_{ii'}(n)$ $(n \geqslant 1)$ as the conditional probability that the service of a packet requires n roundtrip periods (or n transmission attempts) and that the channel state is i' at the end of the service *given* that the service time starts in state i $(i, i' = 0, 1)$, i.e.

$$\gamma_{ii'}(n) \;=\; \text{Prob}[\, n \text{ roundtrips}, r_{k+n(s+1)-1} = i' \mid r_{k-1} = i \,]\,, \tag{168}$$

if the packet is first transmitted in slot k.

Since we assume that the service periods are not interrupted by idle periods, every service period starts in the same channel state as the previous one ended with, which clearly is also the channel state during the *departure slot* of the previous packet. As such, the departure slots mark the boundaries between two successive service periods while the length of a service depends on the channel state during the preceding departure slot, in view of (168). Therefore, the channel state during departure slots forms what is called a (special) Semi-Markov Process, see also the footnote on p. 132. This *embedded* channel state process is of particular importance to us, since we need to know the equilibrium probabilities of being in either channel state when a service period starts (or ends), or equivalently, the channel state *at departure slots* only. Let r_k^* be the channel state during the k-th departure slot and $\pi_{i,k}$ the probability that $r_k^* = i$. Then we have for the row vector $\boldsymbol{\pi}_k$ with elements $\pi_{0,k}$ and $\pi_{1,k}$ respectively,

$$\boldsymbol{\pi}_{k+1} = \boldsymbol{\pi}_k \sum_{n=1}^{\infty} \boldsymbol{\gamma}(n) \qquad \text{with} \qquad \boldsymbol{\gamma}(n) = \begin{bmatrix} \gamma_{00}(n) & \gamma_{01}(n) \\ \gamma_{10}(n) & \gamma_{11}(n) \end{bmatrix}, \tag{169}$$

which is to be compared with (140) for the channel state probabilities $\boldsymbol{\omega}_k$ at consecutive slots. The row vector of equilibrium probabilities of $\{r_k^*\}$ is $\boldsymbol{\pi} = \lim_{k\to\infty} \boldsymbol{\pi}_k$ which is different from the probabilities $\boldsymbol{\omega}$ in (142), as will be shown.

The probabilities $\boldsymbol{\gamma}(n)$ can be found as follows. According to (140), the $(s+1)$-*step transition probabilities* of the channel state process $\{r_k\}$ are given by the matrix $\mathbf{q}^{s+1}$, such that $[\mathbf{q}^{s+1}]_{ii'}$ is the probability that the channel state is i' at the end of a roundtrip period given that it is i at the end of the previous roundtrip. This matrix has eigenvalues 1 (since it is stochastic) and ϕ. The spectral decomposition representation (see Appendix 3.A) for $\mathbf{q}^h$ is given by

$$\mathbf{q}^h = \begin{bmatrix} \bar{\sigma} & \sigma \\ \bar{\sigma} & \sigma \end{bmatrix} + \phi^h \begin{bmatrix} \sigma & -\sigma \\ -\bar{\sigma} & \bar{\sigma} \end{bmatrix} \qquad , h \geqslant 0\,. \tag{170}$$

Now, the matrix $\boldsymbol{\gamma}(n)$ in (169) is found as

$$\boldsymbol{\gamma}(n) = \big(\mathbf{q}^{s+1}\mathbf{e}\big)^{n-1}\mathbf{q}^{s+1}\bar{\mathbf{e}}\,, \qquad n \geqslant 1\,, \tag{171}$$

where the channel error probabilities are arranged in the matrices

$$\mathbf{e} = \begin{bmatrix} e_0 & 0 \\ 0 & e_1 \end{bmatrix} \qquad \text{and} \qquad \bar{\mathbf{e}} = \begin{bmatrix} \bar{e}_0 & 0 \\ 0 & \bar{e}_1 \end{bmatrix}.$$

The geometric-like expression (171) can easily be interpreted as follows. If the service of a packet requires n roundtrip periods, then there are first $n-1$ roundtrip periods (each of length $s+1$) wherein an error occurred followed by one roundtrip without channel error.

The z-transform of the matrix $\boldsymbol{\gamma}(n)$ gives us the probability generating matrix (or *pgm*) $\mathbf{g}(z)$ of the number of roundtrip periods required for the successful transmission of a packet, accounting for the channel state at the start and end of the service. From (171), we find

$$\mathbf{g}(z) \;=\; \sum_{n=1}^{\infty} \boldsymbol{\gamma}(n)\, z^n \;=\; (\mathbf{I} - z\,\mathbf{q}^{s+1}\mathbf{e})^{-1}\mathbf{q}^{s+1}\bar{\mathbf{e}}z\,, \tag{172}$$

where $\mathbf{I}$ is the 2×2 identity matrix. Note that the inverse matrix in (172) always exists for $|z| \leqslant 1$, in the first place because the elements of $\mathbf{g}(z)$ are (partial) pgfs and must be analytic in that region. Another reason why $\mathbf{I} - z\,\mathbf{q}^{s+1}\mathbf{e}$ is nonsingular for $|z| \leqslant 1$ is due to the Perron-Frobenius (PF) Theorem, see [94, 141]. The λ eigenvalues of the matrix $\mathbf{q}^{s+1}\mathbf{e}$ are found as the roots of its characteristic polynomial $\det(\lambda\mathbf{I} - \mathbf{q}^{s+1}\mathbf{e})$ and must necessarily be smaller than one since the matrix is substochastic (except perhaps for some boundary values of $\mathbf{q}$ and $\mathbf{e}$ for which the channel is static instead of correlated). Let us define

$$\begin{aligned} \nu(z) \;&= \det(\mathbf{I} - z\,\mathbf{q}^{s+1}\mathbf{e}) \\ &= 1 - z\big(e_0(\bar{\sigma}+\phi^{s+1}\sigma) + e_1(\sigma+\phi^{s+1}\bar{\sigma})\big) + z^2 e_0 e_1 \phi^{s+1}\,, \end{aligned} \tag{173}$$

then it is clear that the eigenvalues λ satisfy $\nu(\lambda^{-1})=0$ and are equal to the inverse of the roots of $\nu(z)$. These roots, for which the matrix $\mathbf{I}$–$z\,\mathbf{q}^{s+1}\mathbf{e}$ is singular, are therefore situated *outside* the unit disc. Now note that the inverse of a matrix $\mathbf{A}$ is given by its adjoint matrix $\mathrm{adj}(\mathbf{A})$ divided by its determinant $\det(\mathbf{A})$, again see [94, 141]. Hence, we find from (172) the following expression for the pgm $\mathbf{g}(z)$:

$$\mathbf{g}(z) = \frac{z}{\nu(z)}\Big(\mathbf{q}^{s+1}\bar{\mathbf{e}} - z\,\phi^{s+1}\begin{bmatrix} \bar{e}_0 e_1 & 0 \\ 0 & e_0\bar{e}_1 \end{bmatrix}\Big)\,. \tag{174}$$

Recall also that the element $[\mathbf{g}(z)]_{ii'}$ is the partial pgf of the number of roundtrip periods required to successfully transmit a packet ending the service in state i', given that it starts in state i. For future purposes, we also introduce the pgm $\mathbf{S}(z)$ as

$$\mathbf{S}(z) \triangleq \mathbf{g}(z^{s+1}), \tag{175}$$

which gives the distributions of the number of *slots* in a service period rather than the number of roundtrips. The nice thing about the matrix representation $\mathbf{S}(z)$ is that it allows us to handle the distribution of contiguous service times *as if they were independent*. Indeed, the conditional distribution of the length of n contiguous services (i.e. without idle periods in between) ending in state i', given that the first service starts in state i is simply given by $[\mathbf{S}(z)^n]_{ii'}$.

From (168), (169) and (172), the transition probability matrix of the embedded process $\{r_k^*\}$ is seen to be given by $\mathbf{g}(1)$ and is easily obtained from (174) as

$$\mathbf{g}(1) = \begin{bmatrix} q_0^* & \bar{q}_0^* \\ \bar{q}_1^* & q_1^* \end{bmatrix} \qquad \text{with} \qquad q_i^* = \frac{\bar{e}_i}{\nu(1)}\big(\bar{\sigma} + \phi^{s+1}(\sigma - e_{\bar{i}})\big).$$

Hence, the equilibrium probabilities $\boldsymbol{\pi}$ of the channel state r^* at an arbitrary departure slot must satisfy $\boldsymbol{\pi} = \boldsymbol{\pi}\mathbf{g}(1)$, from which

$$\boldsymbol{\pi} = \begin{bmatrix}\pi_0 & \pi_1\end{bmatrix} = \begin{bmatrix} \dfrac{\bar{e}_0\bar{\sigma}}{\bar{e}_0\bar{\sigma} + \bar{e}_1\sigma} & \dfrac{\bar{e}_1\sigma}{\bar{e}_0\bar{\sigma} + \bar{e}_1\sigma} \end{bmatrix}. \tag{176}$$

Remember that for the queue working under overload conditions we have defined the throughput η as the inverse of the mean service time. The distributions of the service times conditioned on the channel state in which the service starts are given by (175), whereas $\boldsymbol{\pi}$ in (176) are the equilibrium probabilities of being in either state 0 or 1 at the start of a service time. Therefore, using the moment generating property of pgfs, we find the throughput as

$$\eta^{-1} = \boldsymbol{\pi}\,\mathbf{S}'(1)\,\mathbf{1} = (s+1)\boldsymbol{\pi}\,\mathbf{g}'(1)\,\mathbf{1} \quad \text{with} \quad \mathbf{1} \triangleq \begin{bmatrix}1\\1\end{bmatrix}, \tag{177}$$

where $\mathbf{1}$ is the 2×1 column vector with 1 on both entries and where $\mathbf{S}'(1)$ indicates the matrix $\mathbf{S}(z)$ with each element differentiated to z and evaluated for $z=1$ (and likewise for $\mathbf{g}'(1)$). After properly evaluating (177), we finally find the following expression for the throughput of the Stop-and-Wait ARQ protocol over the correlated error channel:

$$\eta = \bar{\sigma}\frac{\bar{e}_0}{s+1} + \sigma\frac{\bar{e}_1}{s+1} = \frac{1-(\bar{\sigma}e_0+\sigma e_1)}{s+1}. \tag{178}$$

Note that this expression is surprisingly simple and is equal to that of the throughput for Stop-and-Wait ARQ with an *uncorrelated* error channel, see e.g. [79,146,151,187],

$$\eta_{\text{static}} = \frac{\bar{e}}{s+1},$$

which we would get if the channel has a static error probability $e \triangleq \bar{\sigma}e_0+\sigma e_1$. Equivalently, another way to interpret (178) is to say that the throughput η for our two-state

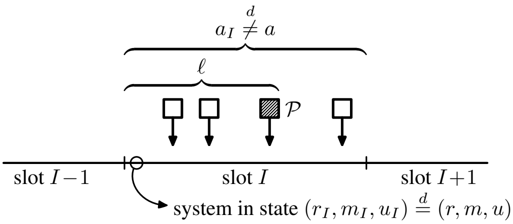


Figure 3.9: The variable ℓ denotes the number of arrivals in slot I that are stored in the queue before $\mathcal{P}$, including $\mathcal{P}$ itself.

modulated channel is simply the weighted sum

$$\eta = \bar{\sigma}\eta_0 + \sigma\eta_1 , \qquad \text{with} \quad \eta_i = \frac{\bar{e}_i}{s+1} \quad , i=0,1 , \tag{179}$$

of the 'static' throughputs which would be obtained in the (fictitious) case of a static channel error probability e_i. Or in other words, η_i is the throughput in case the channel has error probability e_i in *every* slot. The coefficients in (179) are equal to the equilibrium channel state probabilities $\boldsymbol{\omega}$ of (142), which is consistent with the findings in [189]. Observe also that η is independent of K, i.e. as far as the mean service time of an arbitrary packet is concerned, it does not matter how often the channel changes state as long as the overall ratio of 0- and 1-slots remains the same. Unlike the throughput however, the probability $p_0{+}p_1$ of an empty queue *does* decrease with K. See Sec. 3.7 for further discussion.

3.6 Analysis of the packet delay

In this section, we derive an expression for the pgf $D(z)$ of the total delay d experienced by an arbitrary packet traversing the transmitter queue. Of all the packets arriving to the system, consider an arbitrary packet and tag it as packet $\mathcal{P}$. Also, let us mark the arrival slot of $\mathcal{P}$ as slot I. We define the delay d as the number of slots between the end of the slot in which $\mathcal{P}$ arrives (slot I) and the end of the slot in which $\mathcal{P}$ departs from the transmitter queue.

We can quantify the delay as the amount of time required to process the unfinished work present in the queue directly *after* the arrival of $\mathcal{P}$, if we assume for a moment that the arrivals during a slot occur one by one. Clearly, this unfinished work foremost depends on the system state (r_I, m_I, u_I) at the beginning of slot I. Hence, we first need an expression for the system state distribution $P_I(x, y, z)$ in slot I. Now, it has been argued before (see e.g. [43]) that due to the uncorrelated (iid) nature of the packet arrival process, the system as 'seen' by an *arbitrary arriving packet* at the beginning of a slot has the same distribution as the system in an *arbitrary slot*. Therefore, we can immediately use the result (163) of Sec. 3.4 for the joint pgf $P(x, y, z)$ as it is also the distribution of the system state during the arrival slot of an arbitrary packet,

$$(r_I, m_I, u_I) \overset{d}{=} (r, m, u) , \qquad P_I(x, y, z) = P(x, y, z) . \tag{180}$$

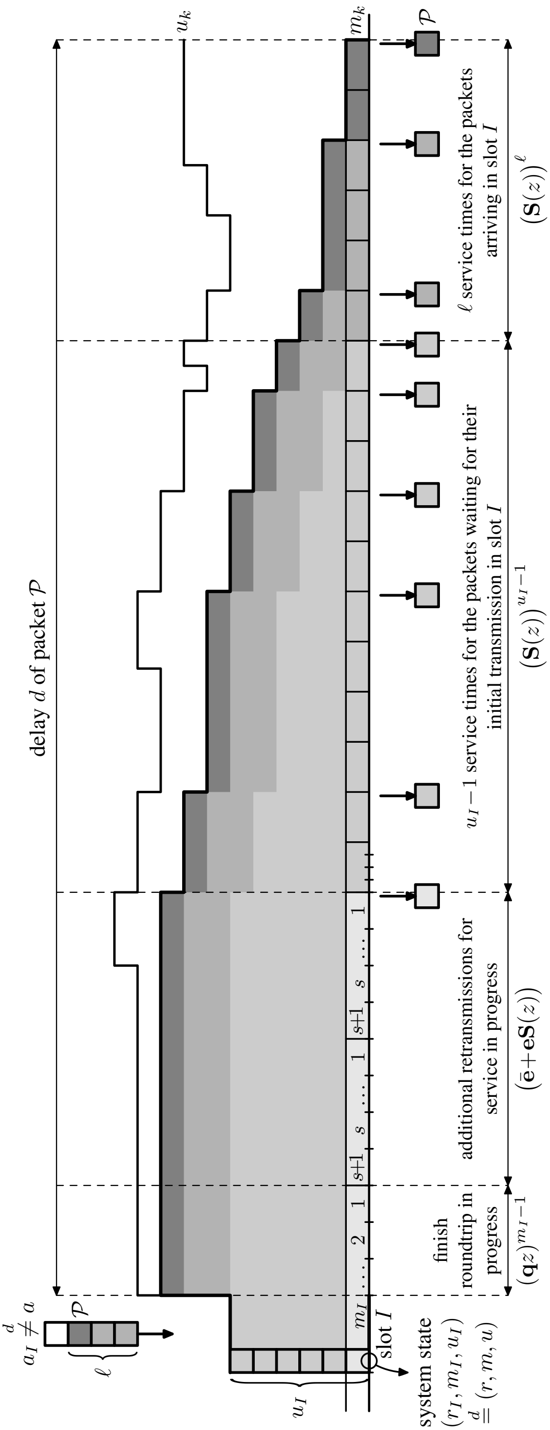


Figure 3.10: The delay d of the arbitrary packet $\mathcal{P}$ that arrives in slot I with system state (r_I, m_I, u_I).

Apart from the system state at the beginning of slot I, the delay of $\mathcal{P}$ also depends on the number and order of arrivals *during* that slot. Let ℓ be the number of packets arriving in the queue in slot I that will be served no later than (but including) $\mathcal{P}$, as indicated in Figs. 3.9 and 3.10. The pgf $L(z)$ of ℓ is found as (see [43])

$$L(z) \;=\; \frac{z(1-A(z))}{\alpha(1-z)}\,, \tag{181}$$

by considering the fact that the pgf of the number of arrivals a_I in slot I is *not* $A(z)$, but rather $zA'(z)/\alpha$ and that $\mathcal{P}$ could be *any* of the a_I arrivals with equal probability. In the following, we derive the pgf $D(z)$ of the delay of $\mathcal{P}$ by conditioning on the system state at the beginning of slot I. Specifically, as in Sec. 3.4, we make the distinction between the cases where the system is *idle* in slot I (i.e. $m_I = 0$) or *busy* ($m_I > 0$). In both cases, we can refer to Fig. 3.10 for a visual representation of the time periods that constitute the total delay of $\mathcal{P}$.

Let us first consider the case where $\mathcal{P}$ arrives when the queue is idle. This means that the first of the ℓ packets will immediately be transmitted in the next slot. If the service period of the first packet is finished, the next of the ℓ packets is served, without interruption in between, and so forth until finally, the packet $\mathcal{P}$ is served. Therefore, we know from the previous section that the pgfs of the length of these ℓ contiguous services ending in channel state i' given that they start in state i ($i, i' = 0, 1$) are the entries of the pgm

$$\mathbf{D}_{m_I=0}(z) \;=\; \sum_{v=1}^{\infty} \mathbf{S}(z)^v \,\mathrm{Prob}[\ell\!=\!v] \;\triangleq\; L(\mathbf{S}(z))\,, \tag{182}$$

where the notation $L(\mathbf{S}(z))$ stands for a matrix that is a power series in the matrix $\mathbf{S}(z)$ with the same coefficients as the power series expansion in z of $L(z)$ given by (181). As argued in (180), the probabilities that the queue is empty and in either channel state during slot I, are given by the vector $\mathbf{p}$ in (156). Hence,

$$\mathrm{E}[z^d\{m_I\!=\!0\}] \;=\; \mathbf{p}\,\mathbf{D}_{m_I=0}(z)\,\mathbf{1} \;=\; \mathbf{p}\,L(\mathbf{S}(z))\mathbf{1}\,. \tag{183}$$

In Appendix 3.A, we derive the spectral decomposition (214) of the matrix $\mathbf{S}(z)$ with eigenvalue functions $\lambda_1(z)$ and $\lambda_2(z)$. Let us agree that the index j is always used to indicate one of the two eigenvalues (i.e. $j = 1, 2$) and that we may write λ_j when in fact, we mean $\lambda_j(z)$. We find

$$\mathrm{E}[z^d\{m_I\!=\!0\}] \;=\; \sum_{j=1}^{2} L(\lambda_j)\,\mathbf{p}\,\mathbf{S}_j(z)\mathbf{1}\,, \tag{184}$$

where $\mathbf{S}_1(z)$ and $\mathbf{S}_2(z)$ are the so-called spectral projectors of $\mathbf{S}(z)$ that are derived explicitly in Appendix 3.A, see (212), (213).

Secondly, we consider the more complicated case when $\mathcal{P}$ arrives when the queue is busy serving another packet. Suppose the system state at the beginning of slot I is $(r_I, m_I, u_I) = (i, h, n)$. From Fig. 3.10 it is seen that the pgfs of the delay period ending in channel state i' given that it starts in state i (i.e. $r_I = i$) $(i, i' = 0, 1)$ are the entries of the pgm

$$\mathbf{D}_{m_I>0}(h,n,z) \;=\; (\mathbf{q}z)^{h-1} \cdot \big(\bar{\mathbf{e}} + \mathbf{e}\,\mathbf{S}(z)\big) \cdot \mathbf{S}(z)^{n-1} \cdot L(\mathbf{S}(z))\,, \tag{185}$$

where each factor corresponds to a certain part of the delay. The first factor indicates the number of slots needed to finish the roundtrip period of the packet being served during slot I. If this roundtrip period is finished, either an ACK or NACK was returned to the transmitter. In case of an ACK, no channel error occurred (probabilities $\bar{\mathbf{e}}$) which means the service is finished and the packet departs from the queue. In case of a NACK, a channel error occurred (probabilities $\mathbf{e}$) and the packet is retransmitted such that an additional *remaining* service time must be accounted for. Note that the conditional length of this remaining service time has a distribution that is also given by $\mathbf{S}(z)$. This explains the second factor $\bar{\mathbf{e}} + \mathbf{e}\,\mathbf{S}(z)$. The third factor is due to the service times of the $u_I - 1$ packets waiting in the queue at the beginning of slot I but that were not yet transmitted then. Finally, the fourth factor accounts for the packets arriving during slot I that are transmitted *before* $\mathcal{P}$ and for $\mathcal{P}$ itself, similar to (182). Now, for $1 < h \leqslant s+1$ and $n \geqslant 1$, let

$$\boldsymbol{\chi}(h,n) \triangleq \big[\text{Prob}[r_I = 0, m_I = h, u_I = n] \quad \text{Prob}[r_I = 1, m_I = h, u_I = n]\big]\;,$$

then by using (185) we find for the distribution of the delay d in case the queue is busy during slot I:

$$\begin{aligned}
\mathrm{E}[z^d\{m_I > 0\}] &= \sum_{h=1}^{s+1}\sum_{n=1}^{\infty} \boldsymbol{\chi}(h,n)\,\mathbf{D}_{m_I>0}(h,n,z)\mathbf{1} && (186)\\
&= \sum_{h=1}^{s+1}\sum_{n=1}^{\infty} \boldsymbol{\chi}(h,n)(\mathbf{q}z)^{h-1}\sum_{j=1}^{2}(\bar{\mathbf{e}}+\mathbf{e}\lambda_j)\lambda_j^{n-1}L(\lambda_j)\mathbf{S}_j(z)\mathbf{1}\\
&= \sum_{j=1}^{2}\sum_{h=1}^{s+1}\sum_{n=1}^{\infty} L(\lambda_j)\Big[z^{h-1}\lambda_j^{n-1}\boldsymbol{\chi}(h,n)\begin{bmatrix}\bar{\sigma} & \sigma\\ \bar{\sigma} & \sigma\end{bmatrix}(\bar{\mathbf{e}}+\mathbf{e}\lambda_j)\mathbf{S}_j(z)\mathbf{1}\\
&\qquad + (\phi z)^{h-1}\lambda_j^{n-1}\boldsymbol{\chi}(h,n)\begin{bmatrix}\sigma & -\sigma\\ -\bar{\sigma} & \bar{\sigma}\end{bmatrix}(\bar{\mathbf{e}}+\mathbf{e}\lambda_j)\mathbf{S}_j(z)\mathbf{1}\Big]\,.
\end{aligned}$$

Note that we have used the spectral decomposition (214) again applied to $\mathbf{D}_{m_I>0}(h,$ $n,\, z)$ in the second line and the representation (170) for $\mathbf{q}^{h-1}$ in the third. From (150) and (156) it is clear that the pgfs of the entries of $\boldsymbol{\chi}(h,n)$ are given by $H_0(y,z)$ and $H_1(y,z)$ respectively, which were determined in Sec. 3.4. Hence, we find for the partial pgf of the delay if the packet $\mathcal{P}$ arrives in a busy slot:

$$\mathrm{E}[z^d\{m_I > 0\}] \;=\; \sum_{j=1}^{2} L(\lambda_j)\,\mathbf{C}_j(z)(\bar{\mathbf{e}} + \mathbf{e}\lambda_j)\mathbf{S}_j(z)\mathbf{1}\,, \tag{187}$$

where we have used the row vectors $\mathbf{C}_j(z)$ defined as

$$\mathbf{C}_j(z) \;\triangleq\; \mathbf{H}(z,\lambda_j)\begin{bmatrix}\bar{\sigma} & \sigma\\ \bar{\sigma} & \sigma\end{bmatrix} + \mathbf{H}(\phi z,\lambda_j)\begin{bmatrix}\sigma & -\sigma\\ -\bar{\sigma} & \bar{\sigma}\end{bmatrix}\,. \tag{188}$$

The entries $C_{0j}(z)$ and $C_{1j}(z)$ follow from (158) by applying the appropriate substitutions for z and y:

$$\begin{aligned} C_{ij}(z) &= \frac{A(\lambda_j)}{\bar{\phi}\lambda_j(z-A(\lambda_j))}\Big[\big((\bar{q}_{\bar{i}} + \phi^{s+1}\bar{q}_i)z^{s+1}\bar{d}_i(\lambda_j) - \bar{\phi}\lambda_j\big)R_i(\lambda_j) \\ &\qquad + \bar{q}_{\bar{i}}(1-\phi^{s+1})z^{s+1}\bar{d}_{\bar{i}}(\lambda_j)R_{\bar{i}}(\lambda_j)\Big] \\ &+ \frac{-z^{s+1}\bar{\phi}^{-1}}{\lambda_j(z-A(\lambda_j))}\Big[(1-A(\lambda_j))\bar{q}_{\bar{i}}(p_0+p_1) + \phi^s(1-\phi A(\lambda_j))(\bar{q}_i p_i - \bar{q}_{\bar{i}} p_{\bar{i}})\Big], \end{aligned} \tag{189}$$

in which everything is already known from Secs. 3.3 and 3.4. After substitution of expression (159) for the functions $R_i(z)$ into (189), we find the more explicit expression

$$\begin{aligned} C_{ij}(z) =& \frac{A^{s+1}(\lambda_j)-z^{s+1}}{\bar{\phi}\lambda_j(z-A(\lambda_j))N(\lambda_j)}\Bigg[\bar{q}_{\bar{i}}A^{s+1}(\lambda_j)\bar{d}_{\bar{i}}(\lambda_j)\lambda_j(1-\phi^{s+1}) \\ &\cdot\big[\bar{q}_i(1-A(\lambda_j))(p_0+p_1) - \phi^s(1-\phi A(\lambda_j))(\bar{q}_i p_i - \bar{q}_{\bar{i}} p_{\bar{i}})\big] \\ &+ \big[N(\lambda_j) - A^{s+1}(\lambda_j)\bar{d}_i(\lambda_j)\big(\bar{d}_{\bar{i}}(\lambda_j)\phi^{s+1}\bar{\phi}A^{s+1}(\lambda_j) - \lambda_j(\bar{q}_{\bar{i}}+\phi^{s+1}\bar{q}_i)\big)\big] \\ &\cdot\big[\bar{q}_{\bar{i}}(1-A(\lambda_j))(p_0+p_1) + \phi^s(1-\phi A(\lambda_j))(\bar{q}_i p_i - \bar{q}_{\bar{i}} p_{\bar{i}})\big]\Bigg]. \end{aligned} \tag{190}$$

Finally, we can bring together (184) and (187) to obtain the *unconditional* pgf $D(z)$ of the packet delay d:

$$\begin{aligned} D(z) &= \mathrm{E}[z^d] = \mathrm{E}[z^d\{m_I=0\}] + \mathrm{E}[z^d\{m_I>0\}] \\ &= \sum_{j=1}^{2} L(\lambda_j)\big[\mathbf{p} + \mathbf{C}_j(z)(\bar{\mathbf{e}} + \mathbf{e}\lambda_j)\big]\mathbf{S}_j(z)\mathbf{1}\,. \end{aligned} \tag{191}$$

This can be simplified further by observing that $\bar{e}_i + e_i\lambda_j$ is in fact $\bar{d}_i(\lambda_j)$ and by using the expression (215) for the vector $\mathbf{S}_j(z)\mathbf{1}$. We find

$$\begin{aligned} D(z) &= \sum_{j=1}^{2}\sum_{i=0}^{1}\frac{1}{2}L(\lambda_j)\big[p_i + \bar{d}_i(\lambda_j)C_{ij}(z)\big] \\ &\qquad\cdot\left[1 \pm \frac{(-1)^i\bar{\phi}\,\mu(z^{s+1}) + 2\bar{e}_{\bar{i}}\bar{q}_i(1-\phi^{s+1})}{\bar{\phi}\sqrt{\psi(z^{s+1})}}\right], \end{aligned} \tag{192}$$

with $\pm$ being $+$ if $j=1$ and $-$ if $j=2$. To summarise, in (192) one has to substitute (190) for $C_{ij}(z)$, (211) for the eigenvalue functions $\lambda_j(z)$, (143) for $\bar{d}_i(z)$, (210) for $\mu(z)$, (209) for $\psi(z)$ and (181) for $L(z)$.

The mass function $d(n) = \mathrm{Prob}[d=n]$ can be obtained from the pgf $D(z)$ by using a numerical inversion algorithm, as will be done for the examples in the next section. Unfortunately, the computational cost of calculating $d(n)$ increases with n such that numerical inversion becomes impractical for large n. Nevertheless, an elegant and accurate approximation for the tail distribution of the packet delay can be found from the asymptotic analysis of $D(z)$, for which we refer to Appendix 3.B. As with expression

(164) for the pgf of the queue content, it is possible to derive the moments of the packet delay from (192) by using the moment generating property of pgfs. For example, we have verified numerically that the mean packet delay $\mathrm{E}[d]$ found as $D'(1)$ is exactly equal to $U'(1)/\alpha$, with $U'(1)$ given by (166), as required by Little's theorem [83].

3.7 Bounds for the probability of an empty queue

The final step in the analysis of Sec. 3.4 was the numerical computation of the last unknown p_0+p_1, being the probability of having an empty system during equilibrium. Nevertheless, it is not always necessary to go through these numerical calculations. In the following, we derive a simple upper and lower bound for $\mathrm{Prob}[u=0]=p_0+p_1$. The upper bound corresponds to the case $K=1$ of an uncorrelated channel and the lower bound is asymptotically correct for $K\to\infty$, i.e. in case the channel is heavily correlated.

3.7.1 No channel correlation ($K=1$)

The analysis of the transmitter queue content for Stop-and-Wait ARQ with an uncorrelated channel and *static* error probability e is given in [187], where the pgf of the transmitter queue content in an arbitrary slot during equilibrium is found to be

$$U_{\text{static}}(z) \;=\; U_{\text{static}}(0)\,\frac{(z-1)\,\bar{e}\,A(z)^{s+1}}{z-A(z)^{s+1}(\bar{e}+ez)}\,. \tag{193}$$

The probability of an empty queue $U_{\text{static}}(0)$ then follows from the normalisation condition $U_{\text{static}}(1)=1$ as

$$\mathrm{Prob}[u_{\text{static}}=0] \;=\; U_{\text{static}}(0) \;=\; 1-\frac{\alpha}{\eta_{\text{static}}} \;=\; 1-\alpha\frac{s+1}{\bar{e}}\,, \tag{194}$$

and the condition (167) for the queue to reach equilibrium is

$$\alpha \;<\; \eta_{\text{static}} \;=\; \frac{\bar{e}}{s+1}\,. \tag{195}$$

Regarding our model of the transmitter queue with a two-stated channel model, it is not difficult to express the probability of an empty queue in the special case that the channel is uncorrelated, i.e. in case $K=1$ (or $\phi=0$). Indeed, if we choose the transition probabilities q_0 and q_1 in Fig. 3.6 such that $q_0+q_1=1$ or equivalently, $K=1$, our model reduces to a simpler model with a *static* channel and the results of [187] can be used. The probability that a packet is transmitted erroneously is then independent from slot to slot and equal to $e=\bar{\sigma}e_0+\sigma e_1$, which is in fact equal to $\boldsymbol{\omega}\,\mathbf{e}\,\mathbf{1}$, the error probabilities in both channel states weighted with the equilibrium probabilities of those states. So the probability of an empty queue follows as

$$\mathrm{Prob}[u=0] \;=\; 1-\alpha\frac{s+1}{1-(\bar{\sigma}e_0+\sigma e_1)} \;=\; 1-\frac{\alpha}{\eta} \;=\; 1-\rho \qquad \text{if } K=1\,, \tag{196}$$

where the load ρ of the system was defined in (167).

3.7.2 Heavy channel correlation ($K \to \infty$)

Secondly, we can also investigate the opposite case in which the correlation of the channel state from slot to slot is very large, i.e. in which asymptotically $K \to \infty$ (or $\phi \to 1$). From our definition of K at the end of Sec. 3.3 we know that if K gets larger, the channel state switches less and less often, resulting in longer 0- and 1-periods of which the mean length increases linearly with K. Note that if K were really *equal* to infinity, the Markov chain in Fig. 3.6 is no longer irreducible and the channel will remain in its initial state forever, such that the channel error probability is the same in all slots, either e_0 or e_1 depending on the starting conditions. For our purposes, it is useful to consider these two extreme cases for $K = \infty$ separately. For instance, suppose the channel state is always i ($i = 0, 1$), then the errors in the channel occur independently from slot to slot and always with the same probability e_i. If the mean arrival rate is not too high, i.e.

$$\alpha < \eta_i \triangleq \bar{e}_i/(s+1)\,, \tag{197}$$

the behaviour of the transmitter queue is the same as in the 'static' case considered in Sec. 3.7.1, only this time with static error probability $e = e_i$. After some time, the queue reaches equilibrium and expression (193) for the distribution of the queue content applies as well as (194) for the probability that the queue is empty, both with $e = e_i$.

For a heavily correlated channel ($K \to \infty$, but not $K = \infty$), the basic observation that we make is the following. Strictly speaking, if the 0- and 1-periods are not infinitely long, we obviously need to perform the analysis as shown in the previous sections of this chapter, complete with the numerical procedure to determine $\text{Prob}[u = 0] = p_0 + p_1$. However, if the 0- and 1-periods are very large, we can still reuse the results from the less complicated analysis with static errors. Suppose we consider a single i-period and focus on the behaviour of the queue during this period only, then we see that the queue is governed by the same system equations as in the model with *static* error probability $e = e_i$. Moreover, since the i-period is also very long and assuming that $\alpha < \eta_i$, we can expect that after some transitional phase a *local* equilibrium sets in. For the remainder of that i-period, the queue then exhibits the same kind of (local) equilibrium behaviour as if the channel were *always* in state i. This local equilibrium ends when the channel switches to state $\bar{i}$. Then, again after some transitional phase, a new local equilibrium may set in if $\alpha < \eta_{\bar{i}}$, and so on.

In the following, we derive an expression for $\text{Prob}[u = 0]$ when the channel is heavily correlated. Without loss of generality, let us assume that $e_0 < e_1$ such that $\eta_1 < \eta < \eta_0$. Obviously, we also assume that *overall* equilibrium can be reached, i.e. that the condition (167) is met. Depending on how high the load α/η of the system is, we can distinguish two cases. In the first case, which we term *symmetric stability*, local equilibrium can occur during both 0- and 1-periods. If local equilibrium only can occur during the 0-periods, then we talk about *compensated stability*.

Symmetric stability ($\alpha < \eta_1 < \eta < \eta_0$):

In this case, the mean arrival rate α is low enough such that both $\alpha < \eta_0$ and $\alpha < \eta_1$. Therefore, we can safely assume that for the larger part of both 0- and 1-periods, the

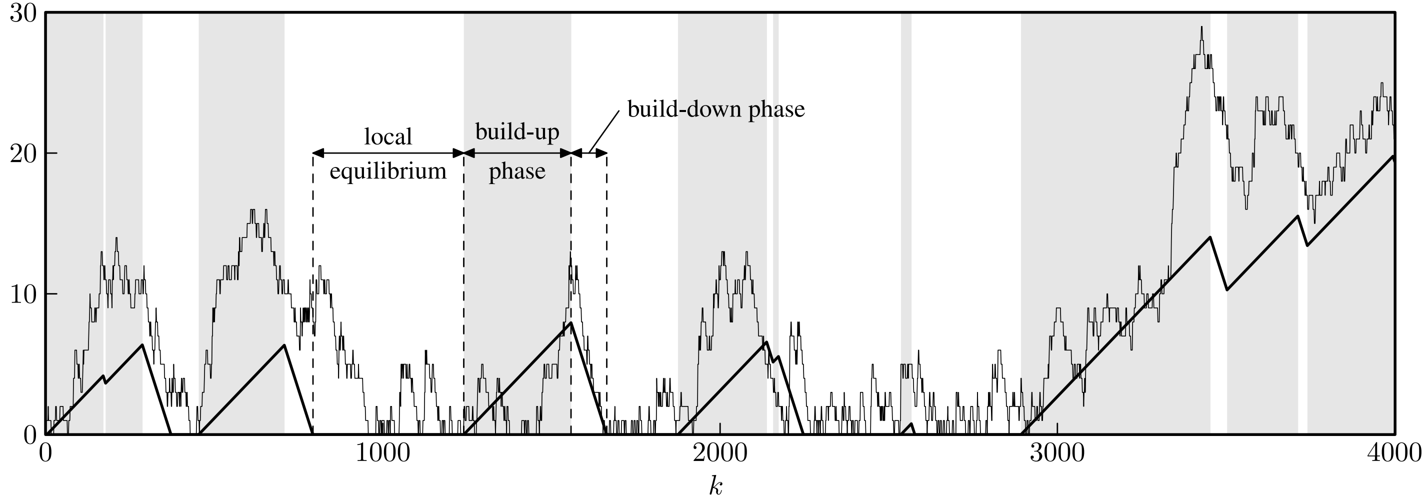


Figure 3.11: Evolution of the content in the transmitter queue in case of *compensated stability*. A trace over 4000 slots is shown for the parameters $s = 3$, $e_0 = 0.1$ and $e_1 = 0.5$. The correlation in the channel has parameters $\sigma = 0.5$ and $K = 100$. The 1-periods (bad channel state) are indicated by a grey background. The bold straight lines indicate the mean excess work Q_k.

queue operates under local equilibrium. In fact, since the periods are very long, we can ignore the transitional phases at the beginning of each period and assume that there is local equilibrium during *all* the slots. Therefore, given that a certain slot belongs to an i-period ($i=0,1$), the probability that the queue is empty during that slot is given by (194) with $e=e_i$. Since a slot belongs to a 0- or 1-period with probabilities $\bar{\sigma}$ and σ respectively, we find

$$\begin{aligned} \text{Prob}[u=0] &= \bar{\sigma}(1-\frac{\alpha}{\eta_0})+\sigma(1-\frac{\alpha}{\eta_1}) \\ &= 1-\alpha\big(\bar{\sigma}\frac{s+1}{\bar{e}_0}+\sigma\frac{s+1}{\bar{e}_1}\big)\,, \qquad \text{if } K\to\infty\,. \end{aligned} \tag{198}$$

Compensated stability $\quad(\eta_1<\alpha<\eta<\eta_0)$:
Here, unlike the previous case, the mean arrival rate is higher than the maximum throughput η_1 during the 1-periods. Obviously, this means there will be no local equilibrium behaviour during 1-periods, since temporarily more packets are entering the queue than can be processed. Instead, as illustrated in Fig. 3.11 for a specific trace of the queue content, the queue will gradually build up as the 1-period advances. We can quantify the rate at which the queue builds up in a fluid-flow like manner as follows. The transmitter queue temporarily works under overload conditions as considered in Sec. 3.5 and transmits at the maximum rate of η_1 packets per slot. Hence, let us call the build-up rate R_{up} the average number of excess packets arriving per 1-slot, then we have

$$R_{\text{up}} = \alpha-\eta_1 = \alpha-\frac{\bar{e}_1}{s+1}\,,$$

i.e. the mean number of packets that enter the queue minus the mean number of packets that leave the queue per 1-slot. The longer the 1-periods are, the longer the queue builds up at this rate and the higher the observed queue sizes will be. Of course, the queue cannot grow indefinitely, since we know that overall stability is assured by (167). Therefore, the build-up during the 1-periods must be compensated for somehow during the 0-periods. Indeed, still reasoning in fluid-flow, there will be a phase at the start of each 0-period during which the queue *keeps* transmitting at its maximum output rate as it did during the preceding 1-period. However, this output rate is now increased to η_0 instead of η_1 packets per slot such that the excess amount of work built up during the 1-periods can now gradually leave the queue. If the 0-period extends long enough, all the excess work will be gone at a certain point and (after a transitional phase) the local equilibrium sets in again. The rate R_{down} at which the queue is emptied at the beginning of the 0-periods is

$$R_{\text{down}} = \eta_0-\alpha = \frac{\bar{e}_0}{s+1}-\alpha\,,$$

i.e. the maximum output rate during 0-periods minus the mean arrival rate. Incidentally, observe that the overall equilibrium condition (167) is equivalent to

$$\alpha<\eta = \bar{\sigma}\eta_0+\sigma\eta_1 \;\Leftrightarrow\; \sigma\big(\alpha-\eta_1\big)<\bar{\sigma}\big(\eta_0-\alpha\big) \;\Leftrightarrow\; \sigma\,R_{\text{up}}<\bar{\sigma}\,R_{\text{down}}\,,$$

which states that on average, the excess work built up in the queue during 1-periods can always be compensated for during the 0-periods. Let the *mean excess work* Q_k be defined as the expected amount of excess packets present in the queue at the beginning of slot k due to the build-up and build-down mechanism, then we have, assuming that $Q_0 = 0$,

$$Q_{k+1} = \begin{cases} Q_k + R_{\text{up}} & \text{if } r_{k+1} = 1\,; \\ \max(Q_k - R_{\text{down}}, 0) & \text{if } r_{k+1} = 0\,. \end{cases} \tag{199}$$

In Fig. 3.11, during the *build-up* phases, which coincide with the 1-periods, the mean excess work Q_k has positive slope R_{up}, while during the *build-down* phases, Q_k has negative slope $-R_{\text{down}}$. In between these phases, where $Q_k = 0$, are the parts of the 0-periods where local equilibrium occurs.

Before we calculate the probability of an empty queue in case of compensated stability with $K \to \infty$, we make the following assumptions:

- First, observe that we can neglect the possibility that the queue becomes empty during the 1-periods when the queue is in a build-up phase. Likewise, we also ignore this possibility when the queue is building down during the 0-periods. In other words, as a first approximation, we assume that observing an empty queue when the mean excess work Q_k is non-zero, is a very rare event which can be ignored. Note that in Fig. 3.11, this event occasionally occurs, but much less frequently than during the periods in which $Q_k = 0$. Also, it is seen that although the queue *can* be empty when $Q_k > 0$, this is only likely to happen when Q_k is relatively small. Obviously, when K grows larger, the build-up phases will also be longer, such that on the whole, there will be less and less slots in which small non-zero values of Q_k occur. Therefore, this approximation gains accuracy as $K \to \infty$.
- Secondly, we assume that in the periods during which $Q_k = 0$, the queue is in local equilibrium, i.e. we ignore the transitional phases in the same way as we did in the case of symmetric stability.

The probability Prob$[u = 0]$ can now be obtained as follows. Consider N pairs of consecutive 0- and 1-periods, with N large. As the mean lengths of 0- and 1-periods are K/σ and $K/\bar{\sigma}$ respectively, these N pairs take up a total of $NK(\sigma^{-1} + \bar{\sigma}^{-1})$ slots on the average. Let χ be the fraction of slots in which $Q_k = 0$, i.e. in which the mean excess work is not building up or down and where we assume the system is in local equilibrium. The number of build-up slots is given by $NK/\bar{\sigma}$, being the number of 1-slots. Since it takes $R_{\text{up}}/R_{\text{down}}$ 0-slots to compensate for the excess work built up per 1-slot, we find for the ratio χ:

$$\begin{aligned} \chi &= \frac{\#\text{ total slots} - \#\text{ build-up slots} - \#\text{ build-down slots}}{\#\text{ total slots}} \\ &= \frac{NK(\frac{1}{\sigma} + \frac{1}{\bar{\sigma}}) - NK\frac{1}{\bar{\sigma}} - NK\frac{1}{\bar{\sigma}}\frac{R_{\text{up}}}{R_{\text{down}}}}{NK(\frac{1}{\sigma} + \frac{1}{\bar{\sigma}})} = \ldots = \bar{\sigma} - \sigma\frac{R_{\text{up}}}{R_{\text{down}}}\,. \end{aligned} \tag{200}$$

During the slots in which the queue is in local equilibrium, the probability of having an empty queue is given by (194) with $e = e_0$. Therefore, the probability of observing an empty queue finally follows as

$$\text{Prob}[u = 0] = \chi(1 - \frac{\alpha}{\eta_0}) = \Big(\bar{\sigma} - \sigma\frac{(s+1)\alpha - \bar{e}_1}{\bar{e}_0 - (s+1)\alpha}\Big)\Big(1 - \alpha\frac{s+1}{\bar{e}_0}\Big) \tag{201}$$

$$= \frac{1}{\bar{e}_0}\big(\bar{\sigma}\bar{e}_0 + \sigma\bar{e}_1 - (s+1)\alpha\big)\,, \qquad \text{if } K \to \infty\,.$$

Note that both (198) and (201) are linear functions of the mean arrival rate α and that the expressions agree when $\alpha = \eta_1$. To summarise, the probability that the queue is empty in case of a heavily correlated channel with error probabilities $e_0 < e_1$ is

$$\text{Prob}[u=0] = \begin{cases} 1 - \rho(\bar{\sigma}\bar{e}_0 + \sigma\bar{e}_1)\big(\frac{\bar{\sigma}}{\bar{e}_0} + \frac{\sigma}{\bar{e}_1}\big) & \text{if } \rho < \frac{\eta_1}{\eta}\,, \\ (1-\rho)\frac{\bar{\sigma}\bar{e}_0 + \sigma\bar{e}_1}{\bar{e}_0} & \text{if } \rho > \frac{\eta_1}{\eta}\,, \end{cases} \tag{202}$$

where we have rewritten (198) and (201) as a function of the load ρ instead of the mean arrival rate α, such that the parameter s no longer appears in the expressions. If the load crosses the threshold value η_1/η, the system switches from symmetric stability to compensated stability.

3.7.3 Upper and lower bound

In the following, let us only consider *positively* correlated channels, i.e. with $K > 1$ or equivalently, $0 < \phi < 1$, which is most often the case in practice. Considering a large set of numerical examples, we found that for K ranging from 1 to ∞ and keeping all other parameters constant, expression (196) is an *upper bound* for the probability $\text{Prob}[u = 0] = p_0 + p_1$ as obtained from the numerical procedure in Sec. 3.4.3. Conversely, expression (202) is a *lower bound*. In other words, for Stop-and-Wait ARQ over a positively correlated Gilbert-Elliot channel, the probability that the transmitter queue is empty is always *lower* than in case of equivalent operation over an uncorrelated channel (with $K = 1$), but higher than when operating over a heavily correlated channel (with $K \to \infty$).

This conclusion is substantiated by the following numerical example. In the first plot of Fig. 3.12, we show the probability of an empty queue versus the system load $\rho = \alpha/\eta$, both in case the channel is uncorrelated ($K = 1$, solid line) and in case it is heavily correlated ($K \to \infty$, dashed line), as was discussed in Secs. 3.7.1 and 3.7.2 respectively. The other parameters of the channel model were chosen to be $e_0 = 0.1$, $e_1 = 0.7$ and $\sigma = 0.5$. The variation of ρ along the abscissa is due only to changing the mean arrival rate α. As is clear from (196) and (202), both curves are (piecewise) linear in ρ. In this plot, we also indicated the regions of stability. For a load ρ lower than the threshold $\eta_1/\eta = 0.5$, the queue operates under symmetric stability, whereas if the load exceeds this threshold, it operates under compensated stability. Obviously, if $\rho > 1$, the queue can no longer reach long-term stability. For three specific values of the load, we show the dependency of $\text{Prob}[u=0]$ on the value of K in the second plot of Fig. 3.12. In each case, this probability is plotted for feedback delays $s = 3, 15, 63$ and for K ranging from 1 to 10^5 on a logarithmic scale. We observe that the curves do indeed stay between the bounds set by the values obtained in the uncorrelated and heavily correlated cases respectively. In fact, the probability gradually decreases in a monotonic way from the upper bound for $K = 1$ to the lower bound for $K \to \infty$, whereby the value of s acts as a kind of scaling parameter.

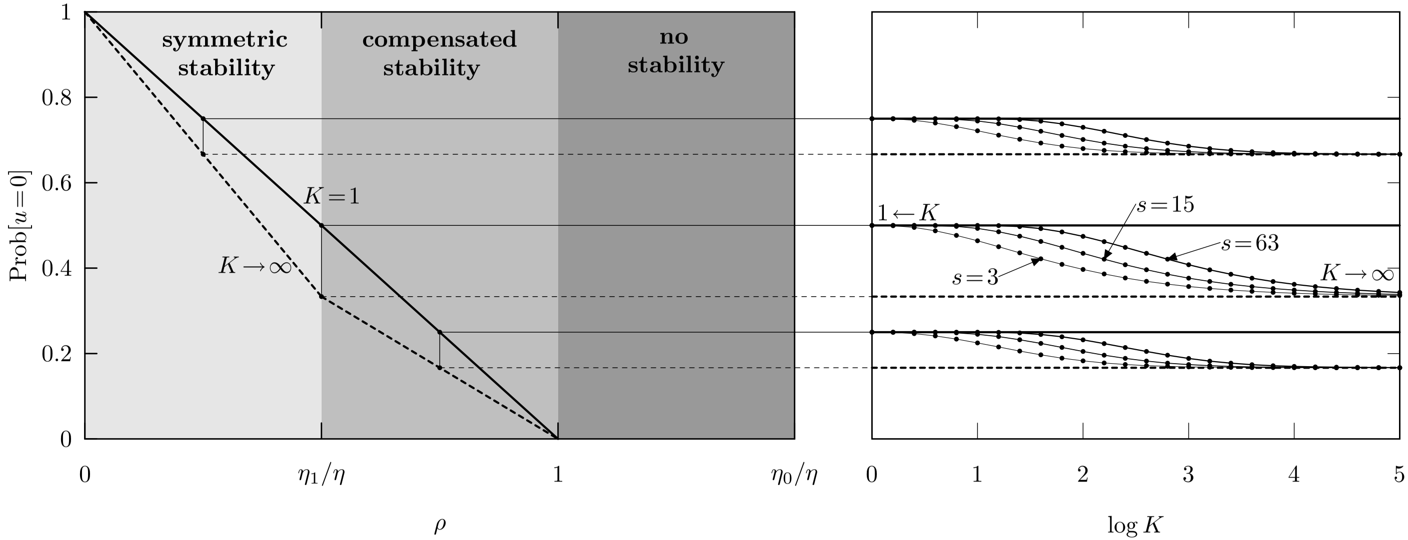


Figure 3.12: On the left, we show $\mathrm{Prob}[u=0]$ as a function of the load $\rho=\alpha/\eta$ both for $K=1$ (solid line) and $K\rightarrow\infty$ (dashed line) in case $e_0=0.1$, $e_1=0.7$, $\sigma=0.5$ and the arrivals per slot are geometrically distributed. For three different values of the load $\rho=0.25, 0.5, 0.75$, the dependency of the *actual* probability of an empty queue on the precise value of K is shown in the right plot. Here, K varies from 1 to 10^5 on a logarithmic scale and three different values of the feedback delay s were considered.

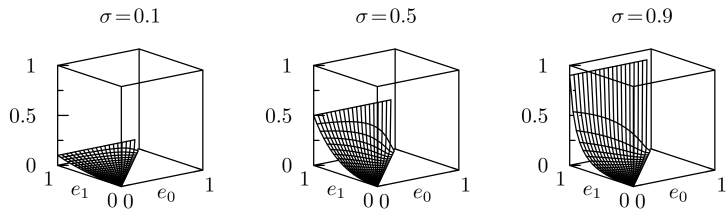


Figure 3.13: Worst-case difference between the upper and lower bound for Prob$[u=0]$ as a function of the error probabilities $0<e_0<e_1<1$, in case of $\sigma=0.1, 0.5, 0.9$ respectively.

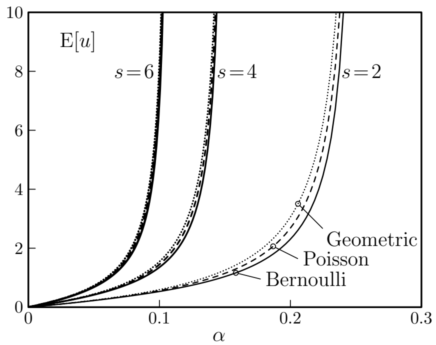


Figure 3.14: The mean queue content versus the mean arrival rate for three different arrival distributions and various feedback delays ($s=2,4,6$).

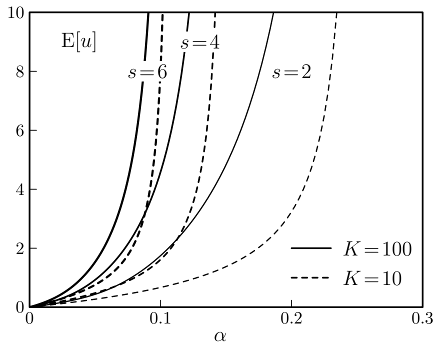


Figure 3.15: The mean queue content for three different feedback delays and for two correlation factors (Bernoulli, $\sigma=0.2$).

The question is now, is there any indication of how close the upper and lower bound lie together? If the bounds are close, they provide a good estimate for the important quantity Prob$[u=0]$ such that the numerical computation of this probability may no longer be required. As can be seen from Fig. 3.12 the bounds are furthest apart when the load is equal to the threshold value η_1/η, so the difference in that point between the upper and lower bound yields a worst-case measure for the distance between the bounds. Taking the difference of (196) and (202) evaluated for $\rho=\eta_1/\eta$, we find

$$\text{worst-case difference} = \sigma\bar{\sigma}\frac{(e_1-e_0)^2}{\bar{e}_0(\bar{\sigma}\bar{e}_0+\sigma\bar{e}_1)}\,. \tag{203}$$

In Fig. 3.13 this measure is plotted versus the error probabilities $e_0 < e_1$ and for $\sigma=0.1, 0.5, 0.9$. It is maximally σ in case $e_1=1$ and becomes very small as the error probabilities are closer together. Note that for values of the load $0<\rho<1$ *other* than the threshold value η_1/η the difference between upper and lower bound will even be smaller than this worst-case measure.

3.8 Numerical examples

In order to illustrate how the equilibrium behaviour of the queue is influenced by the parameters of the model, we now consider some practical examples. First consider the following pgfs for the number of arrivals per slot:

$$\begin{aligned} A_1(z) &= \alpha\, z + 1 - \alpha\,, & A_2(z) &= \exp(\alpha(z-1))\,, \\ A_3(z) &= \frac{1}{1+\alpha-\alpha\, z}\,, & A_4(z) &= \frac{0.5}{1+\alpha_1-\alpha_1\, z} + \frac{0.5}{1+\alpha_2-\alpha_2\, z}\,, \end{aligned} \tag{204}$$

i.e. a Bernoulli, Poisson, geometric and mixed geometric distribution respectively. For the latter, we choose $\alpha_1 = 0.1\alpha$ and $\alpha_2 = 1.9\alpha$.

Equilibrium distribution of the queue content

Let us now concentrate on the mean value $\mathrm{E}[u]$ and the tail distribution $\mathrm{Prob}[u = n]$ of the queue content. As in all further plots in this paragraph, we choose the channel error probabilities to be $e_0 = 0.1$ and $e_1 = 0.8$. For Fig. 3.14 in particular, the probability of being in the BAD state is $\sigma = 0.2$ and the slot-to-slot correlation is $K = 2$. The plot shows the mean queue content $\mathrm{E}[u]$ as given by (166), for different feedback delays $(s = 2, 4, 6)$ and for three different arrival distributions: Bernoulli, Poisson and geometric ones. As one could expect, the number of packets residing in the queue is growing as the feedback delay s is getting larger. Three groups of curves converge to three different asymptotes as for every individual group (corresponding to $s = 2$, 4 and 6), the throughput given by (179) has a different value (around 0.25, 0.15 and 0.1 respectively). Within these groups, the distinction between the curves is due only to the linear contribution of $A''(1)$ in (166), which corresponds directly to the variance of the arrival distribution. As the geometric distribution has the highest variance (and Bernoulli the lowest), it is expected that it will yield the highest (and respectively the lowest) values for $\mathrm{E}[u]$. This can indeed be observed from the plot.

To show the impact of the error correlation in the transmission channel on $\mathrm{E}[u]$, we choose much higher values of K. The results are given in Fig. 3.15, where $\mathrm{E}[u]$ is plotted again for $s = 2$, 4 and 6 but now for $K = 10$ and 100. As for all further plots in this paragraph, we take $\sigma = 0.2$ and the Bernoulli arrival distribution $A_1(z)$. It is clearly seen that the mean queue contents is growing rapidly as K jumps from value 10 to 100. This phenomenon is observed even more distinctly when the feedback delay is relatively short. In order to further investigate this observation, a more detailed analysis for a specific feedback delay $(s = 2)$ was done for $K = 1$, 2, 5, 10, 50 and 100. The results are shown in Figs. 3.16 and 3.17. Fig. 3.16 shows $\mathrm{E}[u]$ plotted against the system load (i.e. $\rho = \alpha/\eta$), whereas Fig. 3.17 is a logarithmic plot showing the tail distribution $\mathrm{Prob}[u > n]$ of the queue content. In Fig. 3.16, some results obtained from simulations according to the described model are included as well.

Equilibrium distribution of the delay

Let us now focus on the distribution of the packet delay d. The Figs. 3.18 to 3.22 are plots of the mass function $d(n)$ of the packet delay. These probabilities were obtained

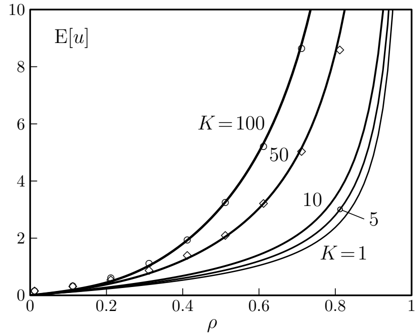


Figure 3.16: The influence of the correlation factor K on the mean queue content (Bernoulli arrivals, $\sigma = 0.2$, $s = 2$).

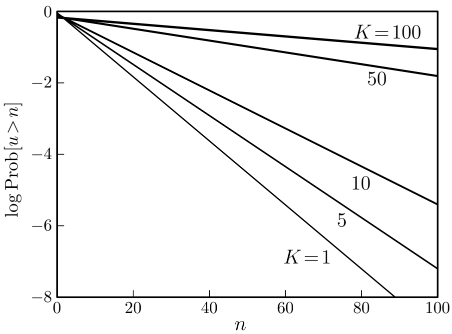


Figure 3.17: Tail distribution of the queue content Prob$[u > n]$ for various correlation factors K and load 0.9 (Bernoulli, $\sigma = 0.2$, $s = 2$).

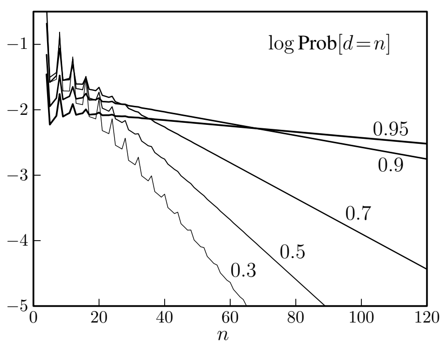


Figure 3.18: Logarithmic plot of Prob$[d = n]$ versus n in case $e_0 = 0.1$, $e_1 = 0.5$, $\sigma = 0.5$ and $K = 10$ for the channel model, feedback delay $s = 3$ and arrivals with geometric distribution, for various values of the load $\rho = 0.3, 0.5, 0.7, 0.9, 0.95$.

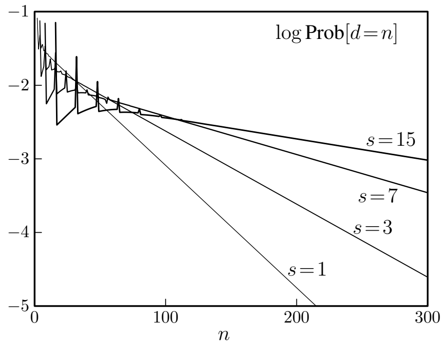


Figure 3.19: Logarithmic plot of Prob$[d = n]$ versus n in case $e_0 = 0.1$, $e_1 = 0.5$, $\sigma = 0.5$ and $K = 10$ for the channel and Poisson arrivals with load $\rho = 0.9$, for various values of the feedback delay $s = 1, 3, 7, 15$.

by numerical inversion of the pgf $D(z)$ given by (192) using the algorithm presented in [20]. For a random variable with pgf $F(z)$, the inversion formula for z-transforms is

$$f(n) = \frac{1}{2\pi \jmath} \oint_{C_r} F(z) z^{-n-1} \mathrm{d}z \,, \tag{205}$$

where $\jmath$ is the complex imaginary unit $\sqrt{-1}$ and C_r is a circular contour around the origin with radius $0 < r < 1$. In [20], this integral is approximated by sampling the integrand on $2n$ points of the contour C_r. Using the discrete Poisson summation formula, the error bound for this approximation is shown to be r^{2n} for large n and small r, such that any desired accuracy is guaranteed by choosing r sufficiently small.

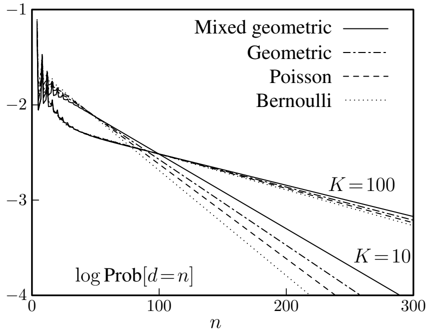


Figure 3.20: Logarithmic plot of $\mathrm{Prob}[d=n]$ versus n in case $e_0 = 0.1$, $e_1 = 0.5$, $\sigma = 0.5$ and feedback delay $s = 3$. For $K = 10$ and $K = 100$ the distribution is plotted for four different arrival distributions $A(z)$: Bernoulli, Poisson, geometric and mixed geometric, each time with load $\rho = 0.9$.

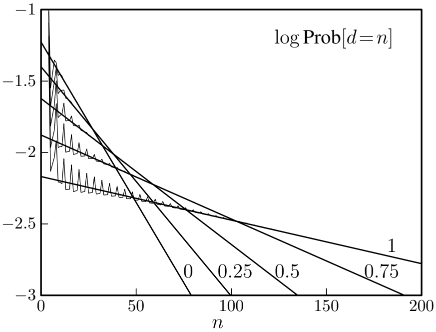


Figure 3.21: Logarithmic plot of $\mathrm{Prob}[d=n]$ versus n in case $e_0 = 0$, $\sigma = 0.5$, $K = 10$, $s = 3$ and Poisson arrivals with load $\rho = 0.9$. The error probability in the BAD state takes values $e_1 = 0, 0.25, 0.5, 0.75, 1$. The bold line indicates the approximated tail distribution.

In Fig. 3.18 we show a logarithmic plot of the mass function $d(n)$ for increasing values of the load $\rho = \alpha/\eta$. The fraction of slots in either channel state is the same ($\sigma = 0.5$), the correlation factor $K = 10$ and the error probabilities in the GOOD and BAD state are $e_0 = 0.1$ and $e_1 = 0.5$ respectively. The feedback delay is $s = 3$ slots and the numbers of arrivals per slot have a geometric distribution. As expected, we observe that the probability mass shifts towards higher values as the load increases. If more packets enter the queue in the same time period, they will have to wait longer before they can be transmitted. Secondly, as with all of the following plots, we see that $d(n) = 0$ for $n < s+1$ since the minimal delay of a packet is one roundtrip period of $s+1$ slots. A packet experiences this minimal delay when it is the first packet arriving during an idle slot and its first transmission is successful. Notice also the peaks in the mass function for n equal to multiples of the roundtrip period $s+1$. The peaks are more pronounced and last into higher values of n when the load is lower, which can be explained as follows. For low load, there is a high probability that packets arrive during a slot when the system is idle. Hence, the first of the packets will be transmitted immediately in the next slot and the delay of all those packets will be a multiple of $s+1$. Note that the probability mass $d(n)$ for n *not* a multiple of $s+1$ is entirely due to arrivals in slots with $m_k = 2, \ldots, s+1$, which is more likely to happen when the load increases. The same observations can be made from Fig. 3.19, where various values of the feedback delay are considered. The arrivals have a Poisson distribution with load $\rho = 0.9$ and the other parameters are as in Fig. 3.18. As one expects, the experienced delay increases with the feedback delay s. In Fig. 3.20 we show the influence of the arrival distribution $A(z)$ for the four different arrival distributions in (204). The system parameters are the same as in Fig. 3.18 and the load is chosen to be $\rho = 0.9$. The plot shows the mass function of the packet delay for these four types of arrivals, both in case $K = 10$ and $K = 100$. The Bernoulli, Poisson, geometric and mixed geometric distributions have variances 0.132, 0.157, 0.182 and

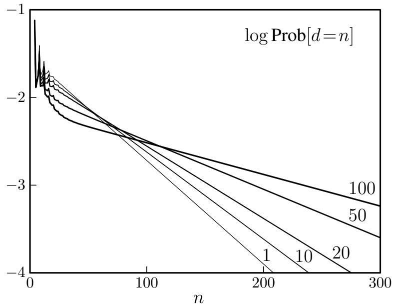


Figure 3.22: Logarithmic plot of Prob$[d=n]$ versus n in case $e_0=0.1$, $e_1=0.5$, $\sigma=0.5$ with feedback delay $s=3$ and Poisson arrivals with load $\rho=0.9$, for various values of $K=1,10,20,50,100$.

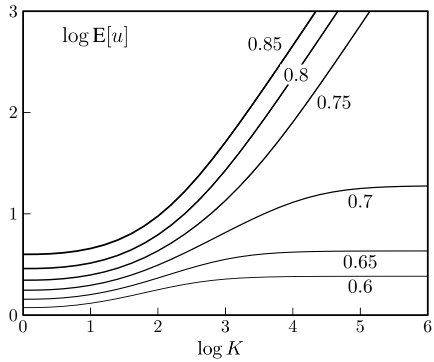


Figure 3.23: Log-log plot of $\mathrm{E}[u]$ versus K in case $e_0=0.1$, $e_1=0.5$, $\sigma=0.5$, $s=3$ and with Poisson arrivals. Curves are shown for system load $\rho=0.6$ to 0.85 respectively. When the load crosses the threshold value $\eta_1/\eta=0.7143$ the queue switches from symmetric stability to compensated stability.

0.222 respectively and we observe that a higher variance of the arrival distribution results in a higher packet delay. In Fig. 3.21, we assume that there are no errors in the GOOD state ($e_0=0$) while we consider increasing values for the error probability in the BAD state, $e_1=0,0.25,0.5,0.75,1$. As expected, the delay increases if the error probability is higher. Additionally, we demonstrate the effectiveness of the dominant pole approximation for the tail distribution of the delay as discussed in Appendix 3.B. The bold lines correspond to the geometric decay of the tail distribution obtained in Appendix 3.B, which gives very good results for high n.

Finally, to illustrate the fact that the correlated nature of the transmission errors in the channel has impact on the delay d as well, we have plotted in Fig. 3.22 the delay distribution for increasing values of the correlation factor $K=1,10,20,50,100$. The arrivals are Poisson with load $\rho=0.9$ for all curves and the other parameters are the same as in Fig. 3.19. Observe that although the fraction of slots in the BAD state is the *same* for all curves, the delay increases drastically with the factor K. The simple fact that *both* the BAD and the GOOD periods *last longer* (see Sec. 3.3) for higher K, results in a higher delay for an arbitrary packet.

Symmetric versus compensated stability

In fact, whether or not the performance *keeps* degrading for increasing correlation in the channel as it does in this example of Fig. 3.22 depends on the *type of stability* the queue is in (see Sec. 3.7.2). Note that in this example the queue operates under compensated stability since the load $\rho=0.9$ is larger than the threshold value $\eta_1/\eta=0.7143$. To illustrate this assertion, we discuss another experiment. In Fig. 3.23 we show a log-log plot of the mean queue content $\mathrm{E}[u]$ as given by (166) versus the correlation factor K. Recall that the mean queue content is closely related to the mean packet delay $\mathrm{E}[d]$ via Little's law. The six curves correspond to increasing values of the load $\rho = 0.6$, 0.65, 0.7, 0.75, 0.8, 0.85. If the load is chosen below the threshold value 0.7143, the queue operates under *symmetric* stability and we see that the curves *saturate* for $K \to \infty$. As such, the effect of a degrading performance for an increasingly correlated channel is somehow limited. However, if the load exceeds the threshold value, the queue switches to *compensated* stability and suddenly the mean queue content no longer saturates but *keeps increasing*, linearly with K. Hence, the performance can become arbitrarily bad.

The reason why this effect occurs can easily be explained when considering the mechanism of queue build-up and build-down of Sec. 3.7.2. Specifically, under compensated stability, the queue content will increase on average during the BAD periods and decrease during the GOOD periods in a way that is described by the *mean excess work* function Q_k given by (199) and illustrated in Fig. 3.11. Now, it is clear that if the BAD periods last longer and longer, the queue builds up uninterruptedly for a longer time and the mean excess work will be allowed to reach higher peaks. Therefore, the excess packets Q_k become an increasingly more dominant part of the total queue content u_k which also grows with the length of the BAD periods. This result strongly shows the importance of accounting for possible correlation in the transmission channel when predicting the queueing performance. Indeed, as we have shown, both queue content and packet delay may be severely underestimated when assuming only *static* errors.

3.9 Conclusion

We have analysed the transmitter queue of the SW-ARQ protocol in case of a two-state Markovian error channel. Closed-form expressions were derived for the pgf of the equilibrium distribution of the system state and the queue content as well as the maximum throughput. By using the eigenvalues and the spectral decomposition of the matrix with the conditional lengths of the service time, we obtained the pgf of the delay experienced by an arbitrary packet. Additionally, we established a useful lower and upper bound on the probability that the queue is empty, thereby distinguishing between operation under symmetric and compensated stability. For the queue content as well as the packet delay, we also gave an accurate approximation for the tail distribution and found that for the latter, the asymptotic contribution of the second eigenvalue function is zero. Finally, by means of some examples we have discussed the influence of the model parameters on the queue performance. The most important observation is that, at least when operating under compensated stability, the performance worsens drastically as the correlation in the channel is higher (i.e. if K increases), although the overall fraction of BAD slots remains the same. This result emphasizes the importance of taking into account the correlation of the errors when dimensioning the buffer space or calculating the packet loss.

3.A Appendix: Spectral decomposition of $\mathbf{S}(z)$

For the analysis of the delay in Sec. 3.6, we need a suitable representation for a function f of the pgm $\mathbf{S}(z)$. The following spectral decomposition theorem allows us to determine $f(\mathbf{S}(z))$ by evaluating f for *scalars* rather than matrices and can be found in any textbook on matrix algebra (such as e.g. [94, 141]).

A square $N\times N$ matrix $\mathbf{A}$ is diagonalisable if there exists a similarity transform $\mathbf{P}$ such that $\mathbf{P}^{-1}\mathbf{A}\mathbf{P}=\mathbf{D}$, with $\mathbf{D}$ a diagonal matrix with entries taken from the spectrum $\sigma(\mathbf{A})$ of $\mathbf{A}$. The spectrum $\sigma(\mathbf{A})=\{\lambda_1,\ldots,\lambda_k\}$ $(k\leqslant N)$ is the set of eigenvalues of $\mathbf{A}$. Let $f(z)$ be a function that is defined on $\sigma(\mathbf{A})$, then there exist matrices $\mathbf{G}_j$ $(1\leqslant j\leqslant k)$ such that

$$f(\mathbf{A}) \;=\; f(\lambda_1)\mathbf{G}_1+f(\lambda_2)\mathbf{G}_2+\ldots+f(\lambda_k)\mathbf{G}_k\,. \tag{206}$$

The matrices $\mathbf{G}_j$ are called the *constituents* or *spectral projectors* of $\mathbf{A}$ belonging to the eigenvalue λ_j and have the following properties:

- $\mathbf{G}_j$ is idempotent, i.e. $\mathbf{G}_j^2=\mathbf{G}_j$.
- $\mathbf{G}_1+\mathbf{G}_2+\ldots+\mathbf{G}_k=\mathbf{I}$, with $\mathbf{I}$ the $N\times N$ identity matrix.
- $\mathbf{G}_j\mathbf{G}_{j'}=\mathbf{0}$ whenever $j\neq j'$ $(1\leqslant j,j'\leqslant k)$.

The projectors $\mathbf{G}_i$ can be obtained by using the Lagrange interpolation formula on $\sigma(\mathbf{A})$:

$$\mathbf{G}_j \;=\; \prod_{j'=1,j'\neq j}^{k}(\mathbf{A}-\lambda_{j'}\mathbf{I}) \Bigg/ \prod_{j'=1,j'\neq j}^{k}(\lambda_j-\lambda_{j'})\,. \tag{207}$$

The extension of this theorem to non-diagonalisable matrices $\mathbf{A}$ also exists but is slightly more complicated.

We can use the above to obtain the spectral decomposition of the pgm $\mathbf{g}(z)$ given by (174). First, let us denote the two eigenvalues of $\mathbf{g}(z)$ as $\lambda_1^*(z)$ and $\lambda_2^*(z)$ respectively, which are the solutions of the characteristic equation $\det(\lambda\mathbf{I}-\mathbf{g}(z))=0$ equivalent to

$$\begin{aligned} 0 \;=&\; \lambda^2\nu(z) \\ &+\lambda z\left[z\phi^{s+1}(e_0\bar{e}_1+\bar{e}_0e_1)-\bar{e}_0(\bar{\sigma}+\phi^{s+1}\sigma)-\bar{e}_1(\sigma+\phi^{s+1}\bar{\sigma})+z^2\bar{e}_0\bar{e}_1\phi^{s+1}\right]. \end{aligned} \tag{208}$$

Let $z\psi(z)$ be the discriminant of this ordinary quadratic equation in λ, i.e.

$$\psi(z) \;\triangleq\; \mu^2(z)+4\bar{e}_0\bar{e}_1\sigma\bar{\sigma}(1-\phi^{s+1})^2\,, \tag{209}$$

with

$$\mu(z) \;\triangleq\; \bar{e}_0(\bar{\sigma}+\phi^{s+1}\sigma)-\bar{e}_1(\sigma+\phi^{s+1}\bar{\sigma})-z\,(e_1-e_0)\phi^{s+1}\,. \tag{210}$$

As in Sec. 3.6, we agree that the index j is always either 1 or 2. Solving the characteristic equation then yields the eigenvalue functions

$$\lambda_j^*(z) \;=\frac{z}{2\nu(z)}\Big[\bar{e}_0(\bar{\sigma}+\phi^{s+1}\sigma)+\bar{e}_1(\sigma+\phi^{s+1}\bar{\sigma}) \tag{211}$$

$$- z\,\phi^{s+1}(\bar{e}_0 e_1 + e_0 \bar{e}_1) \pm \sqrt{\psi(z)}\Big]\,,$$

where $\pm$ means $+$ for $j=1$ and $-$ for $j=2$. It can be verified that if $|z|<1$, the eigenvalue functions $\lambda_j^*(z)$ are situated inside the unit disc too. Therefore, in case $f(z)$ is a pgf and thus analytic for $|z|<1$, the spectral decomposition is always well defined. Also, we have that $|\lambda_1^*(z)| \geqslant |\lambda_2^*(z)|$, i.e. $\lambda_1^*(z)$ is the 'larger' of the two eigenvalue functions and yields the PF-eigenvalue $\lambda_1^*(1)=1$ of the stochastic matrix $\mathbf{g}(1)$ with eigenvector $\boldsymbol{\pi}$ as given by (176). However, although $\lambda_1^*(1)=1$, the function $\lambda_1^*(z)$ is in general *not* a pgf, since its singularities with smallest modulus are not real, as would be required by Pringsheim's theorem (a.k.a. Vivanti's theorem, see [186]). These singularities are the branch points given by the two (complex conjugate) roots of $\psi(z)$. The spectral projectors $\mathbf{G}_j$ of $\mathbf{g}(z)$ are found from (207) and (211) as

$$\mathbf{G}_j(z) \;=\; \pm\frac{1}{2\sqrt{\psi(z)}}\cdot\begin{bmatrix}\mu(z)\pm\sqrt{\psi(z)} & 2\bar{e}_1\sigma(1-\phi^{s+1})\\ 2\bar{e}_0\bar{\sigma}(1-\phi^{s+1}) & -\mu(z)\pm\sqrt{\psi(z)}\end{bmatrix}, \tag{212}$$

with the same convention for $\pm$ as before. Since $\mathbf{S}(z)=\mathbf{g}(z^{s+1})$, we now also have the spectral decomposition of $\mathbf{S}(z)$, with eigenvalues $\lambda_j(z)$ and spectral projectors $\mathbf{S}_j(z)$ given as

$$\lambda_j(z) \;=\; \lambda_j^*(z^{s+1})\,, \qquad \mathbf{S}_j(z) \;=\; \mathbf{G}_j(z^{s+1}). \tag{213}$$

Hence, we can write for the pgm $\mathbf{S}(z)$:

$$f(\mathbf{S}(z)) \;=\; f(\lambda_1(z))\,\mathbf{S}_1(z) + f(\lambda_2(z))\,\mathbf{S}_2(z). \tag{214}$$

From (212) and (213) one also finds for the projectors $\mathbf{S}_j(z)$ postmultiplied by $\mathbf{1}$:

$$\mathbf{S}_j(z)\mathbf{1} \;=\; \frac{1}{2}\mathbf{1} \pm \frac{1}{2\sqrt{\psi(z^{s+1})}}\begin{bmatrix}\mu(z^{s+1}) + 2\bar{e}_1\sigma(1-\phi^{s+1})\\ -\mu(z^{s+1}) + 2\bar{e}_0\bar{\sigma}(1-\phi^{s+1})\end{bmatrix}. \tag{215}$$

We prove the following properties regarding the eigenvalue functions:

Prop. 1: *The function $\lambda_1(z)$ is related to the throughput as*

$$\lambda_1'(1) = \eta^{-1}\,. \tag{216}$$

This follows directly from (211), (213) and (178). □

Prop. 2: *For $j=1,2$, and the function $N(z)$ given by (160):*

$$z - A(\lambda_j(z)) = 0 \;\;\Rightarrow\;\; N(\lambda_j(z)) = 0\,. \tag{217}$$

The eigenvalues $\lambda_j(z)$ are functions for which the characteristic polynomial $\det(\lambda_j(z)\mathbf{I}-\mathbf{S}(z))$, given by (208) with all occurrences of z replaced by z^{s+1}, is zero. Now, substitution of $\lambda_j(z)$ into (160) yields an expression for $N(\lambda_j(z))$ which after some careful rearranging of the terms and substitution of $A(\lambda_j(z))=z$ reduces exactly to the characteristic polynomial of $\mathbf{S}(z)$ in $\lambda_j(z)$. Hence, $N(\lambda_j(z))=0$. □

3.B Appendix: Asymptotic analysis of queue content and delay distribution

We now use the pgfs $U(z)$ and $D(z)$ to assess the asymptotic behaviour of the queue content and delay distribution, i.e. the mass functions $u(n)$ and $d(n)$ for large n.

Let us consider a discrete random variable f with pgf $F(z)$. We use a well-known approximation technique (see e.g. [46]) to obtain the mass function $f(n)$ based on the *dominant singularity* of $F(z)$. Applying the principle of analytic continuation and Cauchy's theorem, it is possible to extend the contour C_r in (205) to contain the whole complex plane ($r \to \infty$) *except* for the singularities of the integrand. Instead of one contour around the origin, there is now a big contour around the z-plane and a number of small contours around the singularities of $F(z)z^{-n-1}$ other than the origin. For large n, the contribution to $f(n)$ of the contour around the singularities with smallest modulus will dominate the contributions of the other contours. Note that since $F(z)$ is analytic within the unit disc, all these singularities have a modulus larger than 1. Both in the case of $U(z)$ and $D(z)$, the dominant singularity is a single pole, which must necessarily be real and positive in order to ensure a nonnegative mass function. Now, with $F(z)$ having a dominant pole z_f of multiplicity one, it follows from the partial fractions expansion of $F(z)$ around z_f that the (dominant) contribution to $f(n)$ of the contour around z_f can be expressed by the following geometric form:

$$\mathrm{Prob}[f = n] \;\cong\; -\theta_f \, z_f^{-n-1} \,, \tag{218}$$

where θ_f is the residue of $F(z)$ in the point $z = z_f$, i.e.

$$\theta_f \;=\; \mathrm{Res}_{z_f} F(z) \;=\; \lim_{z \to z_f} (z - z_f) F(z) \,. \tag{219}$$

In the examples of Sec. 3.8 we have plotted the probabilities $\mathrm{Prob}[f = n]$ on a logarithmic scale, which produces a straight line when n is large as can be seen from (218):

$$\log \mathrm{Prob}[f = n] \;\cong\; \log \frac{-\theta_f}{z_f} \;+\; n \, \log \frac{1}{z_f} \,. \tag{220}$$

The slope of this line is determined by z_f^{-1} which is the geometric *decay rate* of the tail. Both the queue content and the packet delay distribution exhibit this geometrically decaying tail behaviour. All that remains is to determine the values of z_u, θ_u, z_d and θ_d.

3.B.1 Queue content tail distribution

We now determine the dominant pole z_u and the residue θ_u from the queue content distribution $U(z)$. It is seen from (165) that the dominant pole z_u is given by the smallest zero larger than one of the factor $N(z)$. Hence, z_u satisfies

$$N(z_u) \;=\; 0 \,, \tag{221}$$

and can be determined numerically by using e.g. the Newton-Raphson method. To calculate the residue θ_u of $U(z)$ in the point $z = z_u$, we use the expression for $U(z)$

found in (164) and expression (159) for $R_i(z)$. Using de l'Hôpital's rule, this yields

$$\begin{aligned}\theta_u &= \mathrm{Res}_{z_u} U(z) = \lim_{z\to z_u}(z-z_u)U(z) \qquad (222)\\ &= \frac{(1-z_u)A(z_u)^{s+1}}{(1-A(z_u))N'(z_u)}\sum_{i=0}^{1}\bar{e}_i\Big[\phi^s(\phi A(z_u)-1)\big(A(z_u)^{s+1}\bar{d}_{\bar{i}}(z_u)-z_u\big)(\bar{q}_{\bar{i}}p_{\bar{i}}-\bar{q}_i p_i)\\ &\qquad -\bar{q}_{\bar{i}}(A(z_u)-1)\big((\phi A(z_u))^{s+1}\bar{d}_{\bar{i}}(z_u)-z_u\big)(p_0+p_1)\Big]\,.\end{aligned}$$

3.B.2 Delay tail distribution

After close inspection of the factors in (192), we found that the dominant pole z_d of $D(z)$ is always the smallest zero larger than one of the factor $z - A(\lambda_1(z))$ in the denominator of the functions $C_{i1}(z)$ given by (189). It can be shown that the other poles all have a larger modulus. Again, the value of z_d must be obtained numerically. From (217) we observe that z_d is also a zero of the denominator $N(\lambda_1(z))$ of the functions $R_i(\lambda_1(z))$ (see (159)) in $C_{i1}(z)$, i.e.

$$N(\lambda_1(z_d)) \;=\; 0\,. \qquad (223)$$

This would lead us to believe that z_d is a pole of $C_{i1}(z)$ with multiplicity 2. However, after elimination of the functions $R_i(z)$ it becomes clear in (190) that the numerator is *also* zero for $z = z_d$ due to the factor $A^{s+1}(\lambda_1(z)) - z^{s+1}$. The concerning factors in numerator and denominator annihilate for $z \to z_d$ (again using de l'Hôpital's rule) as

$$\lim_{z\to z_d}\frac{A^{s+1}(\lambda_1(z))-z^{s+1}}{z-A(\lambda_j(z))} = -(s+1)z_d^s\,, \qquad (224)$$

and z_d proves to be a pole with single multiplicity due to the factor $N(\lambda_1(z))$ in the denominator of $C_{i1}(z)$.

For convenience, let us define

$$y_d \triangleq \lambda_1(z_d) \quad \text{and} \quad y'_d \triangleq \lambda'_1(z_d)\,, \qquad (225)$$

then, we have from (192):

$$\begin{aligned}\theta_d \;&=\; \mathrm{Res}_{z_d} D(z) = \lim_{z\to z_d}(z-z_d)D(z) \qquad (226)\\ &= \sum_{i=0}^{1}\frac{1}{2}L(y_d)\bar{d}_i(y_d)\mathrm{Res}_{z_d}C_{i1}(z)\left[1+\frac{(-1)^i\bar{\phi}\,\mu(z_d^{s+1})+2\bar{e}_{\bar{i}}\bar{q}_i(1-\phi^{s+1})}{\bar{\phi}\sqrt{\psi(z_d^{s+1})}}\right],\end{aligned}$$

where the residues of $C_{i1}(z)$ in z_d follow from (190) as

$$\begin{aligned}\mathrm{Res}_{z_d} C_{i1}(z) = &\frac{(s+1)z_d^{2s+1}}{\bar{\phi}N'(y_d)y'_d y_d}\Big[\bar{q}_{\bar{i}}\bar{d}_{\bar{i}}(y_d)y_d(1-\phi^{s+1}) \qquad (227)\\ &\cdot\big[-\bar{q}_i(1-z_d)(p_0+p_1)+\phi^s(1-\phi z_d)(\bar{q}_i p_i-\bar{q}_{\bar{i}}p_{\bar{i}})\big]\end{aligned}$$

$$+\,\bar{d}_i(y_d)\big(\bar{d}_{\bar{i}}(y_d)\phi^{s+1}\bar{\phi}z_d^{s+1} - (\bar{q}_{\bar{i}}+\phi^{s+1}\bar{q}_i)y_d\big)$$
$$\cdot\left[\bar{q}_{\bar{i}}(1-z_d)(p_0+p_1) + \phi^s(1-\phi z_d)(\bar{q}_i p_i - \bar{q}_{\bar{i}} p_{\bar{i}})\right]\Bigg].$$

Note that the second eigenvalue function $\lambda_2(z)$ has no contribution to the tail distribution $d(n)$. Finally, from (221) and (223) we find the following interesting relation between the decay rates z_u^{-1} and z_d^{-1} of the queue content distribution and the delay distribution:

$$z_u = \lambda_1(z_d)\,. \tag{228}$$

3.C Appendix: Some literature on ARQ

When modelling transmission protocols over classical *wired* computer networks, it is certainly justified to assume a certain correlation in the channel error process. The 'channel' in this case is a path of wired links between corresponding layers in two nodes of the network. In practical systems, there are various factors that cause the errors in the channel not to occur independently from each other, but in so-called *bursts*. For example, electromagnetic interference with other channels, the unavailability of intermediate network nodes, congestion in these nodes or the presence of traffic with a higher priority are all circumstances that can lead to the loss or erroneous reception of data packets. The impact of these effects is usually variable over time, so that the error probability of subsequent packets during a transmission can hardly be considered as a constant value.

However, if accounting for correlation in the error process during transmission over a fixed network is useful, we can safely say that it is *essential* in case of transmission over a *wireless* medium. Indeed, where errors in a fixed link are mostly caused by external factors, we shall see that for wireless communications, the temporary failing of the channel is *inherent* to the nature of the medium. Physical causes such as the mobility of the users, noise, reflection of the electromagnetic waves on buildings, dispersion and attenuation have as result that the amplitude and phase of the received signal can strongly vary over space and time. The joint result of these effects is known as *channel fading* which has as consequence that during certain periods, the received signal is too weak or too distorted to be decoded. Simple countermeasures such as increasing the channel capacity can only be of limited use here. Increasing the transmitting power of the signal indefinitely is no option either, in view of the limited autonomy of most wireless devices. One can thus safely say that the wireless medium is far less reliable than fixed connections, and protocols at the link layer have to be able to handle the lower quality of the services provided by the physical layer.

So for the performance analysis of a transmission protocol like ARQ over a wireless link, the choice of a suited channel model is very important. If a stochastic model is used for the performance assessment, the results of the analysis are heavily influenced by how we choose to model the error process in the channel. A proper choice for the channel model can be made if we account for the fact that transmission errors of packets are a direct consequence of the channel fading process. For instance, it is clear that the error probability will increase drastically during the periods in which the channel is in deep fading. We thus expect from a representative model for the transmission errors over a wireless channel that it follows the fading process as closely as possible, which means e.g. that it exhibits a similar correlation structure. By allowing a simple form of correlation in the channel model as we did in the current chapter, we move away from the simplifying, unrealistic, but often used assumption that packet errors occur independently from each other. Especially in the case of wireless channels, accounting for the correlated nature of the errors in the channel of the stochastic model is of specific importance for the applicability of the results to real-life situations.

In what follows, we first have a look at the so-called Rayleigh fading process and motivate the choice made in this chapter to modulate the error probability in the channel by a Markov chain with two states. Then we give a survey of some improvements and extensions of the three basic ARQ protocols that have been proposed in literature and finally, we also give an overview of the existing work with regard to the performance analysis of ARQ protocols.

3.C.1 Channel models for wireless communication

Let us again consider the specific situation in which information has to be sent over a single wireless link, from the transmitter to the receiver. At the transmitter side, the information is coded and modulated on a carrier wave, after which the resulting signal is sent into the environment. The receiver then picks up this signal via an antenna, decodes it and delivers the information to the end user. As indicated above, the nature of the medium and more specifically the way in which the signal propagates through the medium will result in a channel fading effect of which the severity varies over time. In general, we can split up the causes of this fading process into large-scale effects and small-scale effects [107, 112, 176]. On a large scale, the primary cause of fading is the fact that propagation through a free medium causes attenuation of the signal strength (*path loss*), proportional to the square of the travelled distance and therefore relatively easy to predict. However, in addition there are often irregularities present on the terrain such as large buildings, billboards or forests that can temporarily block the waves and thus cause variability of the attenuation. If an impenetrable object is situated between the transmitter and the receiver, then the radio signals can often still reach the receiving antenna because of a secondary diffraction field that forms at the edges of the object [164]. This effect is called *shadowing* and is usually modelled by using a lognormal variance around the mean [176]. Also, depending on the size of the textural features on large surfaces, an incoming wave may either reflect or disperse on this surface. In the latter case, the energy is spread out over a large spatial angle. On a smaller scale, about the size of half a wavelength of the carrier wave, fading is also caused by small changes in the distance between the transmitter and the receiver. In mobile communications for instance, the transmitter and receiver are always in relative motion to each other, so the propagation conditions for the signal can vary strongly.

Rayleigh fading

The most dominant form of fading on the small scale is *multipath* fading caused by the fact that a signal usually reaches the receiver along many different paths, due to effects mentioned earlier such as reflection, dispersion and diffraction. Both the resulting attenuation and phase shift of the signal are different on each of these paths. Therefore, it is clear that under certain circumstances, the incoming waves from different directions can interfere destructively at the receiving antenna and thus cause the channel to fade. Mathematically, this phenomenon is described by Clarke's model [58], where it is assumed that the incident energy of the signal is uniformly distributed over all possible directions and that the phase shift is uniformly distributed as well. The complex envelope $g(t)$ of the received signal can thus be written as the superposition of a large number of components,

$$g(t) = \sum_{n=1}^{N} C_n \exp j(2\pi f_D t \cos\alpha_n + \phi_n)\,, \tag{229}$$

where C_n, α_n and ϕ_n respectively are the path gain, the incoming angle and the initial phase shift of the nth component. For large N and for all α_n and ϕ_n distributed independently and uniformly over $[-\pi, \pi]$, this is called *Rayleigh fading* [107, 176, 220]. This is because it can be shown by using the central limit theorem that $|g(t)|$ has a Rayleigh distribution, and that the autocorrelation function (ACF) $f_{|g|}$ is a Bessel function of the first kind of order 0, i.e.

$$\begin{aligned} f_{|g|}(x) &= x\,\exp\Big(-\frac{x^2}{2}\Big), \quad x \geqslant 0\,, \\ R_{gg}(\tau) &= \mathrm{E}[g(t)\,g^*(t+\tau)] \sim 2J_0(2\pi f_D \tau)\,, \end{aligned} \tag{230}$$

which are shown in Fig. 3.24. Under these conditions, the real and imaginary part of $g(t)$ are independent Gaussian processes with mean value 0. A typical example of the evolution of $|g(t)|$ over time is shown in Fig. 3.25. An important parameter in Clarke's model is the maximum Doppler shift f_D, given by

$$f_D \;=\; f_c \frac{v}{c}\,, \tag{231}$$

where v is the maximal relative velocity of the transmitter and the receiver, f_c the carrier wave frequency and c the speed of light. As seen in (229) and (230), $g(t)$ and its ACF only depend on the time t through the factor $f_D t$, so that the *speed* of the fading process is determined by the Doppler shift f_D. In telecommunication applications, the value of f_D lies typically between 5Hz and 300Hz. For example, for a carrier wave of 2GHz and a mobile receiver with a velocity of 108km/h or 30m/s, we have that $f_D = 200$Hz.

The fading speed is of importance when characterising the error behaviour of the channel. Suppose for instance that a sequence of symbols with symbol time T_s (the time required to transmit one symbol) are sent over the channel by the transmitter. The signal of the consecutive symbols[6], during its journey through the medium is being modulated by an envelope $g(t)$ as in Fig. 3.25. If by chance, the channel is in deep fading for the kth symbol, i.e. $|g(kT_s)| < F$ for a certain *fading margin* F, then the received symbol energy will be very low[7]. There is a high probability then

[6]Each symbol consists of a number of bits, depending on the used modulation technique.

[7]A typical value for F is 10dB.

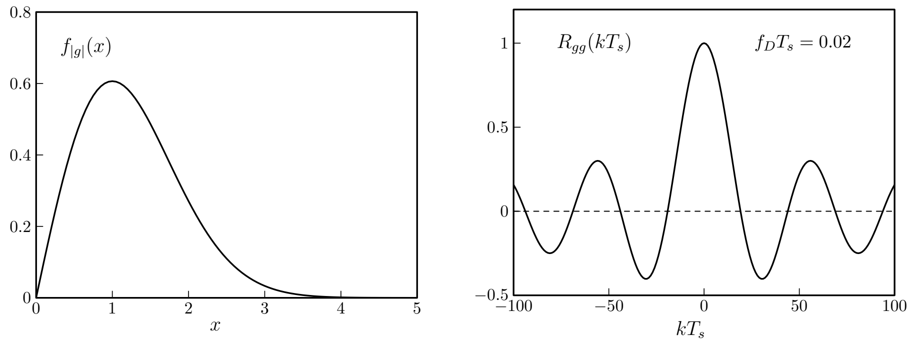


Figure 3.24: The Rayleigh distribution $f_{|g|}(x)$ and the ACF $R_{gg}(\tau)$ of the channel's envelope $g(t)$.

that the SNR (Signal to Noise Ratio) of the received signal is too small to decode correctly and thus that an error occurs. We speak of *slow* fading if $g(t)$ does not change significantly during a large number of subsequent symbol transmissions, which is the case if the 'normalised' Doppler shift f_DT_s is small enough. It is seen that in this case, there will be a substantial amount of correlation in the error process during the symbol transmissions. If the SNR for a certain symbol is high, it will very likely remain high for the next symbols. Conversely, also a period of low SNR (deep fading) will stretch out over many symbol times. So slow fading results in distinct and long periods of high error probability. On the other hand, if f_DT_s is large, we have *fast* fading in which case the channel varies on a timescale smaller than the symbol time T_s. Typically, the channel envelope $|g(t)|$ will then change from a high to a low level and back many times during a single symbol time T_s. An error occurs if the received SNR averaged out over T_s is too small. As a consequence, the occurrences of symbol errors is far less correlated than in the case of slow fading. As such, the error process can sometimes even be represented very accurately by an independent process, i.e. a static (iid) error probability, equal for every symbol [26, 183]. Note that the same conclusion can be drawn from the expression of the autocorrelation function in (230). The higher f_D, the faster the ACF will die away and the lower the correlation at lag $\tau = T_s$. An intermediate situation between slow and fast fading is the case where T_s is of comparable size as the first zero of the ACF in Fig. 3.24. For our analysis of the SW-ARQ protocol, we have assumed that the smallest unit of information is not a separate symbol, but a *packet* of length T_p equal to one slot. Usually, a packet contains multiple symbols, so $T_p > T_s$. The above considerations with regard to the fading speed are valid on the packet level as well, save a rescaling in time. If one is only interested, as we are, in packet errors then the normalised Doppler shift can also be defined as f_DT_p such that the transition from slow to fast fading will be situated around lower values of f_D.

The assumption of a flat (not frequency selective) Rayleigh fading channel is very popular as a general model for the signal propagation through the wireless medium; see e.g. [65, 89, 98, 104, 112, 180, 183, 200, 201, 220, 223, 224] and many others. However, in some situations, other models are better suited. If in (229), there is also a direct

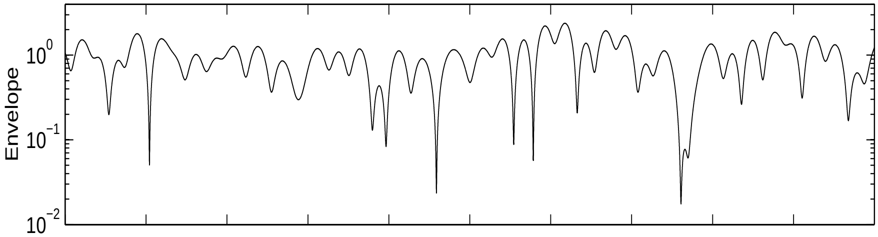


Figure 3.25: An example of the envelope $|g(t)|$ under Rayleigh fading assumptions as a function of time. The mobile velocity is $v = 120$km/h with a carrier frequency of $f_c = 1900$MHz, cfr. [220].

Line-of-Sight (LoS) component without obstructions between the transmitter and receiver, then the mean value of $g(t)$ differs from zero and the expressions (230) are no longer valid. This situation is called Ricean fading [164, 165]. Another and more general model for multipath propagation is Nakagami-m fading [101, 164], which includes Rayleigh fading and also approximates Ricean fading very well.

Jakes' simulator for Rayleigh fading

The mathematical model of Clarke for Rayleigh fading described above was used by Jakes [107] to simulate the fading process. His simulator for the channel propagation is still used very frequently for the evaluation of e.g. coding techniques [127] and transmission protocols [25, 138]. Jakes approximates the envelope in (229) by choosing a finite number of components N and $C_n = 1/\sqrt{N}$, $\alpha_n = 2\pi n/N$, $\phi_n = 0$ for all $n = 1, \ldots, N$. Despite its popularity, this method has some serious shortcomings. For instance, the resulting envelope from Jakes' simulator is clearly deterministic and not stationary in the wide sense [72, 160, 162], which implies among other things that it does not entirely satisfy the desired statistical properties given in (230). Because of this, many improvements on the original simulator have been proposed in literature. Pop *et al.* [162] for example were able to obtain stationarity by representing the initial phase shifts ϕ_n in (229) as independent random variables, uniformly distributed over $[-\pi, \pi]$. Although the ACF of the envelope then converges to $R_{gg}(\tau)$ in (230) for large N, there are still problems with the crosscorrelation between the quadrature components of $g(t)$ and the statistics of higher order. The improved simulator of Zheng *et al.* [220] obtains an exact match with (230) independently of N, by suitably representing C_n, α_n and ϕ_n as independent random variables.

Markovian channel models

During the last decennium, there has been a lot of discussion on the question whether the error behaviour of a Rayleigh fading channel can be adequately modelled by a Markov process, see e.g. [224, 225]. Wang and Chang [201] were the first to give an indication that this is indeed the case, by using an information theoretic argument. Suppose that X_k represents the envelope of the channel when the kth symbol is transmitted, i.e. $X_k = |g(kT_S)|$, then the sequence X_k, $k = 0, 1, 2, \ldots$ forms a Markov process if it satisfies the Markov property from Sec. 1.2.4, i.e. if X_k only depends on the past via X_{k-1}. The dependency on the past can be expressed as follows in terms of the average amount of *information* $I(\cdot)$ or entropy [93, 172]. The amount of 'knowledge' about X_k that is contained in X_{k-1} and X_{k-2} is given by the *mutual* information metric $I(X_k; X_{k-1}, X_{k-2})$, that can be split up as

$$I(X_k; X_{k-1}, X_{k-2}) = I(X_k; X_{k-1}) + I(X_k; X_{k-2}|X_{k-1})\,. \tag{232}$$

The first term is the contribution to the knowledge about X_k contained in X_{k-1} and the second term is the remaining contribution contained in X_{k-2}. It is then clear that the Markov property is satisfied all the better if the second term is small compared to the first one, so if

$$\zeta \triangleq \frac{I(X_k; X_{k-2}|X_{k-1})}{I(X_k; X_{k-1})} \ll 1\,. \tag{233}$$

In [201] it is shown that under Rayleigh fading, ζ is smaller than 0.01 for a normalised Doppler shift below 0.002. So the slower the fading process, the better the Markov property holds, is the conclusion of Wang and Chang. Simulations and experimental data in [89, 180, 200] indeed confirm the applicability of first-order[8] Markov processes as suitable stochastic models for Rayleigh fading. Later, based on a similar argumentation Zorzi *et al.* [223] showed that also the erroneous or correct transmission of packets with length T_p over a Rayleigh fading channel forms a binary Markov chain. In their calculations, it is assumed that if the SNR drops below a certain margin F, the packet is in error or lost. Babich *et al.* [26] consider packet transmissions as well, but use multiple thresholds $0 = F_0 < F_1 < \ldots < F_{N-1} < g_{\max}$ that divide the range $[0, g_{\max}]$ of the channel envelope in N non-overlapping intervals. They research the conditions for which the quantised fading process S_k, where $S_k = F_n$ if $F_n < X_k < F_{n+1}$, is Markovian. For slow fading, the Markov property appears to hold, but only approximately so. For fast fading channels on the other hand, they find the variables S_k to be nearly independent (as we have already explained).

In later studies however, some serious objections are raised to the argumentation and results of Wang and Chang. Tan and Beaulieu [183] for instance, note that a small (conditional) mutual information between X_k and X_{k-2} in the numerator of (233) can also mean that both variables are very *strongly* correlated instead of nearly independent. This is especially the case for slowly fading channels ($f_D T_s < 0.002$) and ζ can therefore be small also if the sequence X_k is not Markov. Secondly, in (232) going back only two steps in time might not be enough. It is possible that the cumulative contribution of information about X_k contained in $X_{k-3}, \ldots, X_{-\infty}$ cannot be ignored, particularly due to the oscillating character of the Bessel function in the ACF of $g(t)$. So instead of taking ζ in (233) as a metric for the Markovian character of the channel, Tan and Beaulieu propose to compare the ACF $R_{gg}(\tau)$ in (230) to the corresponding ACF of the discrete-time Markov chain that is a candidate model for the quantised channel envelope. We mention also the work of Dalke and Hufford [65] who

---

[8] This to say that X_k only depends on X_{k-1} and not on previous states X_j, $j < k-1$, as Wang and Chang want to prove. In this context, for an nth order Markov chain, the state X_k depends on the values of $X_{k-n}, \ldots, X_{k-1}$. Note that a Markov chain of order n with N states can always be reduced formally to a first-order chain by extending the state space to N^n possible states.

show that ζ converges to $1/2$ for slow fading if besides the amplitude of the channel envelope, the phase is taken into account as well, i.e. if $X_k \triangleq g(kT_s)$.

We can conclude from the discussion above that there are theoretical limitations to the applicability of first-order Markov models as a description of the Rayleigh fading wireless channel. Nevertheless, both for simulation and analytical modelling purposes, the use of Markov models to represent the channel behaviour is very common, especially for the performance study of protocols on a higher layer. There are a number of obvious advantages to using Markov models over other kinds of channel models, that explain this success. For instance, if we restrict ourselves to Markov chains with a finite number of states, then a very broad class of stochastic processes[9] can be approximated arbitrarily well, by choosing the number of states large enough and properly fit the transition probabilities. Therefore, the Markov concept is versatile, extendable and usually allows a relatively simple, intuitive analysis of the system involved, which often provides closed-form results for the studied performance measures. Note that extending the state space is a possible solution to the objections raised in [65, 183]. On the other hand, a limitation of Markov models is the explicit geometric decay of their autocorrelation structure, which can make the modelling of phenomena on a large timescale difficult.

The fact that Markov processes conveniently unite both tractability and representativeness for modelling the transmission errors over a communication channel (either wired or wireless) has been recognised very early on. As such, many different kinds of Markov channel models have been proposed, varying in the number of states and the specific error probability in each state as well as the possible transitions between the states. A first model that does better than simply assuming iid errors during the transmission of packets was proposed and analysed by Gilbert [97]. In his model, the channel is always in one of two states: a GOOD state in which a packet is always received correctly, and a BAD state in which an error occurs with probability e_1. The other parameters of the model are the transition probabilities between both states. Later, Elliott [76] assumed a certain error probability e_0 in the GOOD state as well and the resulting two-state channel model is now known as the *Gilbert-Elliott model* (GE). The intuitive relation of this model with the observation that errors occur in bursts is clear if we choose $e_0 \ll e_1$. A burst with a lot of errors in the channel corresponds to a sojourn time in the BAD state, while a period with relatively few errors indicates a stay in the GOOD state. Just like us, many authors use this simple channel model for the performance study of coding schemes [98, 205], ARQ [48, 168, 189, 224] (and many others, see Sec. 3.C.4) and other protocols [75].

Nevertheless, Markov models with *more* than two states are frequently used as well. The Finite-State Markov Channel (FSMC) proposed by Wang and Moayeri [200] models the received SNR under Rayleigh fading conditions and currently is seen as a de facto standard model. As explained on p. 129 with regard to the paper of Babich *et al.* [26], the range of the SNR is quantised into N intervals. The FSMC resides in state n ($1 \leqslant n \leqslant N$) if the SNR lies in the nth interval and only transitions between adjacent states are allowed, which is acceptable for slow fading. The thresholds F_n are carefully chosen in such a way that the equilibrium probabilities $\pi_n = \text{Prob}[S_k = n]$ of all states are the same, i.e. $\pi_1 = \ldots = \pi_N = 1/N$. Bischl and Lutz [32] compare

[9] Except perhaps for self-similar or long range dependent (LRD) processes, see [142].

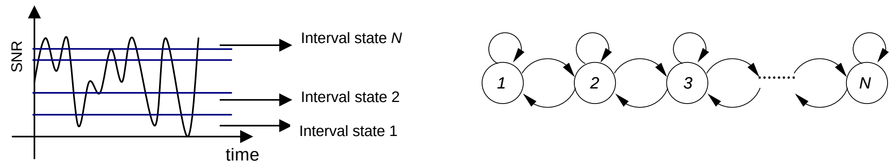


Figure 3.26: The states of an FSMC according to [200] are derived from the received SNR. Transitions are only possible between adjacent states.

the error probability of an arbitrary packet (packet error rate, PER) calculated from an FSMC model with the PER observed for a flat Rayleigh fading channel. Goldsmith and Varaiya [98] study the capacity of a general FSMC (GFSMC) *without* restrictions for the transitions between states. They consider the channel on the timescale T_p of packet transmissions and with a specific error probability in each state. The Rayleigh fading channel is taken as an example here. For a classical type of GFSMCs known as the *Fritchman* models [89], the N states are divided into two classes of respectively GOOD states and BAD states, again without restrictions for the possible transitions. In a BAD state, the error probability is always 1.

If an FSMC or GFSMC, regardless of its structure or complexity is chosen as a channel model, the question obviously remains how to make a proper choice for the parameters of the model. A sensible solution would be to use existing estimation techniques that are used in the framework of hidden Markov models[10] (HMM), see e.g. [95, 174, 180, 190]. In this way, Turin and Van Nobelen [190] obtain the model parameters by observing the channel envelope $|g(t)|$. Swarts and Ferreira [180] use a 'level-crossing' argument, whereby the transition probabilities are estimated as the number of transitions between every possible pair of states during a certain observation time. A drawback of using HMM however, is the lack of an intuitive relation between the channel behaviour and the underlying Markov chain.

In our opinion, the strength of the GE model with only *two* states lies among other things in the fact that with only a small number of parameters, it allows to capture the drastic effect that error correlation in the channel can have on the queueing behaviour of the protocol. Assuming more than two states would undoubtedly describe the channel quantitatively in a more accurate way, but the queueing analysis would have to resort to numerical methods in an early stage such that the results would be less explicit and less elegant than in the case of only two states. Therefore, it would be much more difficult to draw *qualitative* conclusions from the results. In this regard, we can concur with the opinion of Zorzi and Rao [225] that '... *though the error process induced by the Rayleigh fading has nontrivial statistical properties, and the interaction between this process and a queueing system is involved, a surprisingly simple channel model is found to provide very accurate predictions, making analytical computations possible in some cases* ...'. A discussion by the same authors with regard to a suitable choice for the number of states in an FSMC is found in [224], where the GE model is compared to simulations (Jakes' simulator). It appears that unlike the sojourn times in the GOOD state, the distribution of the length of the BAD periods (or the error bursts

[10]For an extensive overview of HMMs in general, see [77].

in the simulation) deviates significantly from the geometric distribution that they are supposed to have according to the GE model. Rather, the mass function of the simulated error burst lengths seems to be the composition of two geometric distributions with different decay (i.e. mixed geometric). Therefore, the authors conclude that a model with *three* states, namely one GOOD and two BAD states, would probably be much more accurate. Interesting as well is [52], where Chen and Rao study the circumstances in which the number of states of an FSMC can be reduced. They demonstrate how a model with N states can be stochastically bounded by a model with only two states. Also Zhang and Kassam [217] wonder how many states an FSMC should have. The received SNR values are partitioned in a number of states according to a criterium based on the mean sojourn time $\bar{\tau}_n$ in every state n, that is chosen as a fixed number of transitions, i.e.

$$\bar{\tau}_n = \bar{\tau} = cT_p = \frac{\pi_n}{C(F_{n-1}) + C(F_n)} .$$

Here, $C(F)$ is the number of times per time unit that the SNR crosses the level F in one direction, which according to Clarke's model is given by

$$C(F_n) = \sqrt{\frac{2\pi F_n}{\gamma_0}} f_D \exp\Big(-\frac{F_n}{\gamma_0}\Big) ,$$

with γ_0 the average received SNR. This yields N equations from which $F_1, \ldots, F_{N-1}$ and c can be calculated for given N. So for every N, the value of c can be obtained, and thus also the mean sojourn time $\bar{\tau}$. For a certain normalised Doppler shift $f_D T_p$, this results in a monotonic decreasing relation $N(\bar{\tau})$ that gives the number of required states for a specific $\bar{\tau}$. The eventual choice of $\bar{\tau}$ and thus of N, is then based on the following considerations. On the one hand, an SNR interval $[F_{n-1}, F_n[$ in Fig. 3.26 must be large enough to contain the SNR variation during *one* packet transmission time T_p. On the other hand, it must also be small enough for the model to be accurate, i.e. the channel quality should not vary too much during a sojourn time in the same state n.

Other channel models

If the length of the error periods or the error-free periods is non-geometric, one can decide not to use a Markov model, but a *semi*-Markov model, see e.g. [206]. For a normal DTMC with N states and transition probabilities $[\boldsymbol{P}]_{ij} = p_{ij}$, $1 \leqslant i, j \leqslant N$, the time between transitions is unspecified in theory (but in practice usually taken to be fixed: one slot, or the time T_p). Semi-Markov processes (SMP) on the other hand, are so-called *doubly* stochastic processes for which the times between the transitions have a specific pmf $f_{ij}(n)$, $n \geqslant 0$ that depends on the states i and j between which the transition is made[11]. The transition matrix $\boldsymbol{P}$ of the 'embedded' DTMC is then a parameter of the model, just like the mass functions $f_{ij}(n)$. Note that in principle, a semi-Markov process can always be approximated by a regular Markov process, see

[11] Semi-Markov processes are also known as Markov renewal processes (MRP). If the distribution of the time between transitions depends only on the *current* state j, i.e. $f_{ij}(n) = f_j(n)$, then this is sometimes called a 'special' semi-Markov process (SSMP) [142].

e.g. [190]. Conceptually, an SMP as a channel model is almost as simple as an FSMC, but often can be made more accurate with regard to the statistical properties of higher order. On the other hand, with an SMP channel model it is usually much harder to conduct the performance study in an analytic way. However, for simulation purposes this is no objection.

The 'bipartite' model of Willig [206] is an SMP with a number of GOOD states and a number of BAD states of which the embedded DTMC is periodic with period 2, i.e.

$$\boldsymbol{P} = \begin{bmatrix} \mathbf{0} & \boldsymbol{Q}_1 \\ \boldsymbol{Q}_2 & \mathbf{0} \end{bmatrix} .$$

The channel always alternates between a GOOD state without errors and a BAD state with iid errors. The distributions of the time between the transitions can be chosen at will. Another SMP model is the McCullough channel model [137], which is in fact a *nonhomogeneous* FSMC, with two transition matrices $\boldsymbol{P}$ and $\boldsymbol{Q}$. If the previous packet was in error, then $\boldsymbol{P}$ is used to make the transition to the next state, and if no error occurred then $\boldsymbol{Q}$ is used. An entirely different approach is proposed by Costamagna *et al.* [60], who use a deterministic channel model based on the chaotic behaviour of two sets of nonlinear equations, that generate error bursts on large and small timescale respectively.

Applications

Aráuz *et al.* [25] and McDougall *et al.* [138] study the transmission of UDP (User Datagram Protocol) traffic[12] over IEEE 802.11 Wireless LAN under Rayleigh fading conditions. Simulations and measurements of the lengths of error bursts and error-free bursts are compared to the GOOD and BAD periods of the GE model. They conclude that in principle, the GE model is inadequate to produce the observed error statistics. Much better agreement is obtained with a 2-state semi-Markov process, where the distributions of the time between transitions are fitted to the measured traces. Qiao *et al.* [163] research the possibilities to enhance the throughput of the IEEE 802.11 protocol by dynamically adapting the PHY mode (the standard provides 8 different PHY modes) to the conditions of the channel. Their method can be applied for any model of the channel variation and specific results are given for a Markov channel model with two states.

Obviously, mobile telephony is an important application as well. A standard of the third generation (3G) for ubiquitous data exchange is for instance UMTS [166, 204] that uses access technology called W-CDMA (Wideband Code Division Multiple Access), to divide the resources of the channel between many simultaneous users. Hueda [104] reports that the packet transmission errors under CDMA can be adequately modelled by an FSMC with two states. A specific method to determine the parameters of the model is given as well. A second generation (2G) standard for mobile telephony is the well-known GSM (Global System for Mobility), that can not only carry voice, but also data. Here, the channel is shared according to a TDMA scheme

[12] UDP is a very rudimentary protocol at the level of the transport layer that does *not* provide any error control or connection mechanisms, this in contrast to its counterpart TCP. In this way, an erroneous or lost UDP packet unambiguously indicates an error in the physical transmission of packets over the channel.

(Time Division Multiple Access). Konrad *et al.* [119] developed a semi-Markov channel model with two states for the transmission of packets via TDMA, based on the analysis of experimental measurements. They compare two FSMC models of first and third order respectively.

Nguyen *et al.* [154] focus on the length of error bursts and error-free bursts measured in UDP traffic over AT&T WaveLAN. They too conclude that the lengths of these bursts are not geometric, but rather have a mixed geometric and a piecewise Pareto distribution instead. Obviously, this compromises the applicability of the GE model for this technology. Dube *et al.* [75] use the GE channel model for the evaluation of the Network File System (NFS) protocol over a wireless link.

3.C.2 Techniques for error control

Traditionally, there are two main error control techniques by which the reliability of information transmission over an error-prone channel can be enhanced. First there is the class of ARQ retransmission techniques, which we will discuss to some extent in this appendix and have already done so in Sec. 3.1. As we have seen, the use of ARQ requires that the channel is bidirectional and that the packets are protected by an error-*detecting* code C_0 [153]. If an error is discovered in a received packet, the receiver simply discards it and requests a retransmission of the same packet.

An entirely different way of coping with transmission errors is FEC (Forward Error Correction) [132] which avoids in advance that the receiver would have to request a retransmission. Obviously the absence of retransmissions has a beneficial effect on the delay of the packets in question. This is achieved by adding a lot more redundant bits to each packet, which enable the receiver not only to detect an error but possibly also to correct it. In other words, the bits are added according to an error-*correcting* code C_1 which allows to recover from a limited distortion in the packet[13]. The more redundant bits are used, the more powerful the error-correcting capabilities of C_1 and the higher the probability that the receiver can correctly reconstruct the packet if its transmission was in error. On the other hand, a large number of extra bits for every packet implies a longer transmission time and thus a lower throughput of the protocol. Therefore it is inefficient to increase the strength of the code indefinitely, so a trade-off has to be made. Note that if the channel is not bidirectional, it is impossible to organise selective retransmissions. In this case, FEC is the only possible means of error control. Benice and Frey [29] discuss the circumstances in which either FEC or a form of ARQ (SW, GBN or SR) is to be preferred, with both reliability and throughput as performance criteria. They conclude that in a channel with independent errors FEC has a higher throughput then ARQ if the error probability is high and vice versa for low error probability. In case the errors occur in a correlated manner, ARQ is almost always better than FEC both in terms of reliability and throughput.

In 1960, Wozencraft and Horstein [209] coined the idea of combining the respective advantages of both FEC and ARQ. This resulted in a whole line of powerful error control schemes now collectively known as *hybrid* ARQ [61, 132, 156]. In these protocols, the packets are coded with an error-correcting code C_1 as is the case for FEC. However, if the number of bit errors introduced by the channel is too large for C_1

[13] An error-correcting code is a fortiori also error-detecting.

to reconstruct a packet, then the receiver can also request the retransmission of that packet (ARQ). So by using a stronger code C_1 instead of C_0, hybrid ARQ reduces the number of required retransmissions at the cost of a longer transmission time T_p for every packet. This way, a significant increase of the throughput can be realised for a broad range of channel conditions.

This initial form of FEC/ARQ combining is also called type-I hybrid ARQ to indicate the difference with the more sophisticated type-II hybrid ARQ [150,216] that was first introduced by Lin *et al.* [131,202]. In these schemes, the redundant parity bits are added to the user information in an *incremental* way, dispersed over multiple packets. In the scheme of Lin *et al.* both C_0 *and* C_1 are used, respectively for the detection of transmission errors and for the generation of redundant bits for error correction. If in a received packet I an error is discovered by C_0, the receiver sends back a NACK and *stores* the erroneous packet in a buffer instead of discarding it. In reaction to the NACK, the transmitter does not simply send I again, but rather $P(I)$ with a (limited) number of extra bits according to code C_1. The receiver then combines the information in both $P(I)$ and the previously stored erroneous packet I and tries to recover the original packet I. If C_1 does not succeed in this effort, or if $P(I)$ itself was transmitted incorrectly, then the receiver can store $P(I)$ along with I and request the transmitter to send even more redundant bits $P'(I)$ (according to an even stronger code C_1'), and so on. Especially because the size of the extra blocks of redundant bits can be kept small, this method can yield a higher throughput than type-I hybrid ARQ. Later, the throughput was increased even further with the discovery of very strong codes like LDPC (Low Density Parity Check) codes or Turbo codes.

The previously discussed channel models have been used for the evaluation of the error control schemes mentioned in this section as well. Nikaein and Bonnet [156] for instance, use an FSMC to study the performance of an adaptive transmission protocol that dynamically switches between normal ARQ and combined FEC/ARQ (type-I) in case the channel conditions are bad. Mukhtar *et al.* [150] present a model for packet errors under type-II hybrid ARQ where a packet is followed through multiple layers of the network: a packet consists of z frames, that contain blocks, which in turn are composed of individual bits. They assume a Gilbert model for the bit errors and eventually arrive at a model with $16z+1$ states for the packet errors.

Note that the performance of FEC and hybrid ARQ error control schemes is mainly measured by the quality of the error-correcting code [213], a topic which is out of our scope. In what follows, we therefore limit ourselves to an overview of studies on 'pure' ARQ protocols where packet errors are mitigated by using *retransmissions* of the same packet rather than using stronger codes.

3.C.3 Performance analysis of ARQ with uncorrelated errors

Although we have clearly motivated above why the transmission errors in a wireless channel should be modelled as a correlated process, it has long been common practice to analyse ARQ protocols using a channel model with *uncorrelated* (or 'static') errors. In what follows, we give a brief summary of some of these analyses for the classical protocols SW, GBN and SR, first in terms of their throughput and then also in terms of the queueing behaviour at the transmitter side. Unless stated otherwise, the mentioned publications treat ARQ models in discrete time, with largely the same assumptions as for the model in this chapter.

Throughput

Very likely, the SR-ARQ protocol was first proposed by Stuart in [179], where the throughput in case of a static error probability is determined. A first comparison of the throughput and the resulting PER (Packet Error rate) for SW, GBN and SR was given by Benice and Frey [28]. Yu and Lin [215] determined a lower bound for the throughput of SR-ARQ in case the resequencing buffer can only contain a limited number of packets and the range of sequence numbers is limited as well. Two numerical methods for the exact calculation were proposed by Jianhua *et al.* [108]. Moris [148] and Turney [192] consider an SW-ARQ scheme whereby a packet is split into n blocks and every block is protected separately by the code C_0. After reception of all the blocks in the packet, the receiver requests the retransmission of the erroneous blocks only. Finally, if all blocks are received correctly, an ACK is sent to the transmitter which then starts transmitting the blocks of a new packet.

Queueing behaviour of the transmitter queue

Saeki and Rubin [170] study the ARQ transmitter queue in case of a TDMA channel. The queue is modelled in discrete time with an iid arrival process, as in this chapter, but for static errors in the channel. They obtain the equilibrium distribution and the moments of the packet delay in case of SW and GBN, as well as a lower and upper bound for the mean packet delay in case of SR. Also interesting is the work of Lu and Chang [135] who determine in a very general manner the pgf of the time required to transmit a batch of M packets over a channel with static errors for both GBN and SR. However, this is no genuine 'queueing' analysis, since they assume the batch always arrives in an empty system and thus that the first packet of the batch can always be transmitted immediately. The delay distribution of the ith packet in the batch is determined as well.

For SW and GBN, Towsley and Wolf [187] calculate the throughput and the pgf of the number of packets in the transmitter queue. Specifically, in case of GBN, they employ a 'modified' model in which a packet leaves the transmitter queue s slots (the time of one feedback delay) earlier than it is supposed to, i.e. immediately after its last (re)transmission instead of the moment when its ACK is returned to the transmitter. In this way, a model is obtained in which every packet has one single, contiguous 'service time', which is easier to analyse. Note that this approach is exact *and* much simpler than the use of Konheim's [118] model for GBN with exponentially growing state space. However, the approach by Towsley and Wolf cannot be applied to SR, whereas that of Konheim can (see next paragraph). Finally, we also mention the work of Fayolle *et al.* [81] who analyse the queue content of the SW-ARQ transmitter in continuous time and with Poisson arrivals. The feedback delay s however, is not fixed for every packet here, but has and iid distribution with maximal value t_{out}: a time-out.

Analysis of the SR-ARQ transmitter buffer

To the best of our knowledge, the only author who has ever presented a queueing analysis of the SR-ARQ transmitter that is both complete and *correct* at the packet level, is Konheim [118]. He proposes a queueing model in discrete time of the transmitter queue and divides the buffer space in a so-called 'transmit' buffer and a 'transit' buffer. The transmit buffer contains the packets that have not yet been transmitted, whereas the transit buffer holds the at most s outstanding packets[14]. The pgfs of both the total number of packets at the transmitter side and the packet delay (without resequencing) are obtained with the same kind of DSVT method as we use in this thesis. Indeed, the analysis is based on an $(s+1)$-dimensional system state description $(u_k, b_{1,k}, \ldots, b_{s,k})$ in slot k, where u_k is the total number of packets in both transmit and transit queue and where $b_{j,k}$ is either 1 or 0, depending on whether a packet was transmitted in slot $k-j$ or not. The analysis is not too difficult, were it not for the fact that a set of linear equations needs to be solved in 2^{s+1} unknowns. As such, this system has a complexity that grows exponentially with the feedback delay s. Konheim analyses GBN in this way as well, in which case the unknowns can be determined in linear time. Also Anagnostou *et al.* [22] give an exact approach to obtain the distribution of the SR-ARQ transmitter queue content, but their analysis is situated entirely in the probability domain. Just like Konheim, they view the system as a multidimensional Markov chain and construct its corresponding transition matrix. Unfortunately, this is where the analysis ends: no solution for the equilibrium distribution of the system state nor for the queue content. Instead, the authors propose to truncate the (extremely large) state space and use numerical methods to compute the solution. A rather crude decoupling approximation is proposed, which is to assume that in each slot there is a fixed probability p that a NACK is returned to the transmitter.

Analysis of the SR-ARQ resequencing buffer

Rosberg and Sidi [167] study a simplified version of SR with regard to the joint distribution of the number of packets in the transmitter and resequencing queue. Bruneel *et al.* [42] obtain expressions for the pgf of the content of the resequencing queue under heavy-traffic conditions. Xia and Tse [210] consider the resequencing phenomenon as well, but not necessarily in the framework of ARQ. They assume that every packet experiences a random delay with iid distribution and conclude from their analysis that the tail decay of this distribution has a severe impact on the performance of the resequencing buffer. Recent work of Rossi *et al.* [168, 169] focusses on the packet delay for SR-ARQ in case of a correlated error channel. However, instead of the delay in the transmitter queue, they concentrate on the transmission time and the resequencing delay at the receiver side. The analysis in [168] assumes a GE channel model under a heavy traffic assumption. In [169], the channel model is specially suited to UMTS W-CDMA.

[14] Note that the transit buffer corresponds to what we have called the 'window' in Sec. 3.1.2.

3.C.4 Performance analysis of ARQ with correlated errors

Throughput

Fujiwara *et al.* [90] determine the throughput for SW and hybrid GBN in case of a GE channel model where errors can only occur in the BAD state ($e_0 = 0$). Leung *et al.* [126] calculate the throughput for GBN as well, but only in case $e_1 = 1$ and $e_0 = 0$. Bruneel and Tison [48] consider the throughput of SW-ARQ over a general GE channel.

Cho and Un [53] and Zorzi and Rao [222] consider the throughput of GBN over a GE channel where errors can occur both in the forward and the feedback channel. The interesting thing about this analysis is the fact that one has to account for some possible events that do not occur for classical GBN. For example, an ACK/NACK message could get lost under these assumptions, so the transmitter has to work with a time-out mechanism. If a valid ACK/NACK message hasn't returned within the time-out period, the packet is considered to be received in error. However, the first authors to consider ARQ with possible errors in the feedback channel were Kim and Un [113] who studied GBN and SR. They assumed two independent GE channel models for the forward and the backward channel respectively. Turin [191] determines the throughput of GBN over a Markovian channel with errors in both directions as well. Interestingly, just as we do in the current chapter, he uses spectral decomposition techniques in the special case that the channel has only two states.

Queue content and delay

It is the work of Towsley *et al.* [189] that probably bears the closest relationship to our analysis of the SW-ARQ model in this chapter. Their previous analysis in [187] of SW and GBN over a channel with static errors is extended here to a GFSMC channel model with N states. The channel has a general transition matrix $\mathbf{P}$ and if the channel is in state n, a transmission error occurs with probability e_n. The throughput is determined for SW and GBN, as well as the pgf of the transmitter queue content for SW and GB(∞). This analysis is therefore a bit more general than ours, but the resulting expressions are less explicit. Also, our approach is different on several accounts and Towsley did not give results for the pgf of the packet delay in the transmitter queue.

In [136], Lu and Chang apply their method of [135] to the analysis of SW, GBN and SR for two different channel models: a Markovian and a Semi-Markovian model. The throughput of the three ARQ protocols is determined as well as the mean packet delay of a batch of M packets that arrives in an empty system. An interesting observation is the fact that for SR, the throughput is independent of the correlation in both channel models.

SR-ARQ with correlated errors: simplifying assumptions

We have already seen that the exact queueing analysis of SR-ARQ is a very difficult task, even in the simple case of static channel errors. If correlation is added to the error process, we can expect the complexity to rise even further. That is why authors who wish to evaluate the performance of SR over wireless channels are often forced to make (sometimes drastically) simplifying assumptions. Kim and Krunz [110] for instance, consider SR-ARQ with both correlated arrivals to the transmitter queue and correlated errors in the channel. Both processes are modelled as Markov processes with respectively N and 2 states (GE channel). In their analysis however, they assume that there is *no feedback delay*, i.e. that it is immediately known at the transmitter side whether a transmitted packet is (or better: will be) received in error or not. This idea is taken from Anagnostou [22], see our discussion on p. 137, and is known as *ideal ARQ*. This approximating assumption relieves us of the need to keep track of the positions of each outstanding packet in the window (or transit buffer), as was required in the analysis by Konheim [118]. Kim and Krunz conduct the analysis in the transformation domain and obtain the pgfs of the queue content and packet delay at the transmitter side, as well as an approximate expression for the mean resequencing time. Fantacci [80] too gives an analysis of SR-ARQ over a GE channel using the same simplifying assumptions. To our knowledge, there are currently no studies available that give an idea of the applicability of ideal ARQ, i.e. how far the approximated results deviate from the exact results under several conditions.

3.C.5 Improved ARQ protocols

Let us consider for now the transmission of packets over a channel with *static* error probability e. The classical ARQ schemes SW, GBN and SR as discussed in Sec. 3.1.2 work fine if the channel has a low error probability e and low feedback delay s, and are therefore suited for many classical applications of data transmission. Especially the continuous schemes SR and GBN are nearly optimal under these conditions in terms of throughput. However, in case of modern non-conventional channels such as wireless or satellite communication where both the error probability and the feedback delay can be very high, the efficiency of these schemes can be improved. Some of the methods that have been proposed are the following:

- *m-copy ARQ, systematically transmitting multiple copies of each packet*

  For high e and s, the throughput can be increased by not transmitting only one copy of a packet each time it is scheduled for transmission or retransmission, but *multiple* copies, immediately after each other. Only after transmitting the mth copy, the transmitter waits for an acknowledgement of this packet. If this is a NACK, then all of the m copies have been received erroneously and a new group of m copies is transmitted. Note that the probability of receiving at least one of the m packets correctly is $1–e^m$, which is higher than $1–e$ if only one packet is transmitted. So the time lost by transmitting each packet m times is compensated by the reduction of the number of required retransmissions, particularly so if e and s are high. If $s<m$ then it is possible that the transmitter receives an ACK for packet i *before* all of the m copies have been transmitted. Naturally, the transmitter no longer continues transmitting packet i then, but proceeds with the next packet in line. Note that the throughput can never be higher than $1/m$ and thus that this method is inadequate if the error probability e is low. A typical threshold value for e above which m-copy ARQ is useful is $e>0.5$.

  Birell [31] analyses the throughput of the method outlined here. Annamalai *et al.* [23] also investigates the optimal m for a given error probability e. In [145], Moeneclaey and Bruneel suggest to take $m = \infty$ which means that for each packet, the transmitter keeps sending copies until an ACK for this packet returns. In [146], the same authors calculate the optimal number of copies m for m-copy SW-ARQ. De Munnynck *et al.* [70] give an analysis of the transmitter queue in case of m-copy GBN-ARQ, with uncorrelated errors in the channel.

Bruneel and Moeneclaey [39] consider the following general model for the number of copies in case of GBN-ARQ. If a packet i is scheduled for its first transmission, it is consecutively transmitted m_0 times. If no ACK has been returned to the receiver for any of these packets s slots later, then the transmitter reacts by transmitting another group of m_1 copies and generally, m_j copies are used for the jth retransmission of the packet. Here also, if $s < m_j$ then it is possible that an ACK is returned for packet i before all m_j copies have been transmitted, after which the transmitter continues with the next packet. The authors also determine the value of m_j as a function of e and s that maximises the throughput. It turns out that the optimal value is independent of the number of retransmissions j.

One can also choose not to send multiple copies immediately, but only if a retransmission is requested, i.e. if a NACK is returned. Sastry [171] and Morris [147] for example, suggest to repeat each retransmission m times in case of SW and to keep sending the concerning packet continuously until an ACK has returned in case of GBN. The 'SETRAN' scheme of Lin and Yu [130] increases the throughput of GBN by letting the receiver decode and acknowledge the packets that are normally ignored by the conventional GBN protocol after an erroneous reception. If a NACK for packet i is returned, the transmitter 'goes back' N slots in time as usual, but now only retransmits the packets that were in error and packet i in the slots that become available this way[15].

The improved SR scheme of Weldon [203] also realises a significant increase of the throughput for high e. If a NACK is returned to the transmitter, then the number of copies that is used for the retransmission of the concerning packet depends both on the occupancy of the resequencing buffer and the number of retransmissions this packet has already been subjected to. The required information is sent back to the receiver along with the acknowledgement messages. Chang and Leung [51] optimise the parameters of Weldon's SR scheme, just like Annamalai *et al.* [23].

► *The use of memory at the receiver side*

Suppose that a packet has been received erroneously for a number of times. Sindhu [173] then proposed, instead of discarding the packet every time, to store the erroneous copy in a *memory* and try to reconstruct the content of the packet by combining the information of the multiple copies that have been received. This idea is based on the observation that in repeatedly transmitted copies of a packet, the bit errors are unlikely to occur at the same positions every time. Obviously, a suitable coding scheme is required to implement ARQ with memory[16]. Bruneel [41] applies the idea to m-copy SW-ARQ with the following conceptual model for the decoding memory at the receiver side. It is assumed that every time a copy of a certain packet is received, the receiver will be unable to reconstruct the correct packet from the

[15] If the receiver does not ignore packets, the acknowledgement messages are indeed returned in time for the receiver to take action according to this rule.

[16] Note that type-II hybrid ARQ could fit into this concept.

copies in memory with a probability that decreases by a factor $\alpha \leqslant 1$ for every additionally received copy. In other words, after receiving j copies, the probability of requiring another retransmission is reduced to $e\alpha^{j-1}$. Distinction is made between FIS (Full Information Strategy) where all copies received thus far are used for decoding and LIS *(Limited Information Strategy)* where only the m copies of the last retransmission are kept in memory. In [185], Tison and Bruneel also calculate the throughput of m-copy SR-ARQ with memory, over an uncorrelated channel. In [48] they do the same for SW-ARQ over a GE channel. De Munnynck *et al.* [71] give the analysis of the transmitter queue and obtain results for the tail distribution of the packet delay, for SW-ARQ and a number of different decoding schemes with memory.

► *Limiting the number of retransmissions (time-out)*
For applications in real-time, such as e.g. multimedia traffic, the data is delay-sensitive (see Sec. 1.1.3). If a packet carrying such data is under way too long, the data becomes useless to the application, so there is no sense in allowing more than a limited number of retransmissions at the level of the link layer. In such situation, a *time-out* period t_{out} can be implemented such that if no ACK is received for a packet within the period t_{out} after its first transmission, the packet is considered to have expired and will not be delivered to the end user.

► *Hierarchical ARQ protocols*
In some cases, ARQ is used in two or more layers of the network. Vitsas and Boucouvalas [195] study the use of ARQ in the IrDA (Advanced Infrared) standard, where the link layer is split up in multiple sub-layers. In the lowest layer (AIr-LC) the GBN protocol is used, whereas in the layer above (AIr-MAC) an optional SW-ARQ mechanism can be used. The authors obtain analytic results for the throughput in case of a static error channel, both with and without the optional SW-ARQ in the upper layer. Their conclusion is that hierarchical ARQ can be beneficial in case of high error rates. Yoshimoto *et al.* [214] consider a similar model with multiple destinations to which messages are sent consisting of several packets. The packets of a message are transmitted using GBN, but for each completed message a separate acknowledgement is required in a SW manner.

Chapter 4

A queue with place reservation

4.1 The model with a single reservation

A common problem in packet-based communication networks such as IP (Internet Protocol) or ATM (Asynchronous Transfer Mode) is the provisioning of adequate QoS (Quality of Service) guarantees to the traffic flows in the network. Moreover, the specific QoS required by a particular traffic flow often depends on the specific application that generates the flow on a higher layer. As indicated in Sec. 1.1.3, we can roughly distinguish two kinds of traffic. For real-time applications such as e.g. video or audio streaming, it is important that the end-to-end delay experienced by the data packets is not too large, i.e. the mean delay and delay jitter should be as low as possible. However, these delay-sensitive applications are generally more forgiving towards packets being timed out or lost in the network. Non-real-time applications on the other hand, such as data transfer, demand as little packet loss as possible but can tolerate much larger delays (loss-sensitive applications).

Much effort has been done to equip the nodes in packet-based networks to acknowledge and support this differentiation in QoS requirements [96]. As such, different scheduling algorithms were suggested and implemented in practice, which provide a better QoS to selected flows than the simple FIFO (First-In First-Out) scheduling where all packets are regarded as equally important.

A theoretically ideal and fair way to share the server capacity over different input streams is Generalised Processor Sharing (GPS) [13, 159, 212], but this mechanism is difficult to apply in packet-based networks, so adaptations for packet scheduling are needed. For instance in the framework of ATM [27], weighted-round-robin (WRR) and weighted-fair-queueing (WFQ) [69] were proposed to reduce the delay of certain flows in the node [134]. For these mechanisms, there are separate queues for each type of traffic and the server 'visits' each queue in a cyclic and weighted manner. Usually, these scheduling mechanisms require knowledge of the traffic mix to function properly and are difficult to implement. To accommodate delay-sensitive traffic, packet-discarding strategies such as push-out buffer (POB), partial buffer sharing (PBS) [55] and random early detection (RED) (see [87, 143]) have been presented and analysed in literature. Also, in IP-based networks, the introduction and enhancement

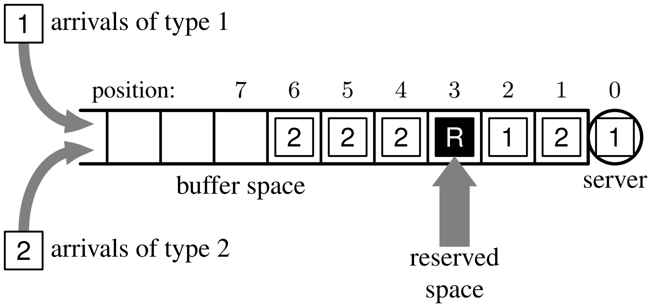


Figure 4.1: Position numbering in the queue with 1 reserved space. State of the system at the begin of slot k is $u_k = 6$, $m_k = 3$. The space was reserved by the 1-packet at position 2.

of the DiffServ [33] and IntServ [37] architecture allows to provide QoS suited to the requirements of specific applications. A simulation study of either is found in [178]. A controllable DiffServ mechanism is Proportional Differentiated Services [73]. Another mechanism that tries to limit the delay of a selected flow is earliest deadline first (EDF) [140, 175] or the use of virtual clocks [218]. An overview can be found in [96, 149].

Suppose we have a queue with two types of packet arrivals and consider the delay experienced by both types of packets. The first arrival flow (type 1) carries delay-sensitive traffic and the second flow (type 2) represents the best-effort traffic, or elastic traffic as it is also called. If the queue operates under the FIFO discipline, no special arrangements are made to prioritise the delay-sensitive flow. Therefore, FIFO may serve as the reference discipline for QoS-unaware network nodes. On the other hand, the most extreme way of priority scheduling is Absolute Priority (AP) or HOL-priority (Head of Line), either preemptive or non-preemptive [54,91,182,196,197,199]. Under this queueing discipline, if the server becomes available and there are type 1 packets present in the queue, a type 1 packet will always be scheduled first, regardless of how many type 2 packets are present and how long they have been waiting for service. This AP discipline was analysed extensively in e.g. [196], from which we draw results for comparison. It is clear that, the server capacity being what it is, AP guarantees the lowest possible delay for the type 1 traffic. However, this comes at the cost of increasing the delay for the packets of type 2. This increase of the delay for the best-effort packets can be very dramatic, especially when the partial load of the prioritised flow is high, and may result in packet starvation or time-out.

In this chapter, we study a new and promising delay priority discipline introduced by Burakowski and Tarasiuk [49], that is simple to implement and which reduces the problem of type 2 packet starvation. The idea is to introduce a reserved space (R) in the queue for future arrivals of type 1, as shown in Fig. 4.1. Whenever a packet of type 1 enters the queue, it takes the position of the reservation that was created there by a previous arrival of type 1 and creates a new reservation at the end of the queue. Type 2 arrivals always take place at the end of the queue in the usual FIFO way. It is seen that this Reservation discipline may allow a 1-packet to jump over some type 2 packets when it is stored in the queue, thus reducing its queueing delay. For instance, with regard to Fig. 4.1, a new arrival can directly jump to position 3 if it is of type 1, instead of having to queue up at position 7 if it is of type 2. Note that it is impossible for any 1-packets to show up behind the reservation, i.e. to have a position number larger than that of R. As long as it is not seized by a 1-packet, the reserved space R behaves as a normal packet in that it shifts one place to the right every time a packet leaves the server. However, the reserved space cannot enter the server at position 0, nor can it leave the queue.

In [50], this idea is carried through further with the proposition of the 'Priority Forcing Scheme' (PFS). In this scheme, a certain application may not only send data packets (D-packets) to the queue, but also reservation packets (R-packets). The R-packets are of small size and require very little service time. Their only purpose is to reserve a space in the node for future arriving D-packets. Evidently, the more R-packets are sent by the application in advance, the higher the possible gain in delay performance for the D-flow. Both in [49] and [50], the reservation mechanism is studied by means of a simple continuous-time model with Poisson input flows. The analysis in those papers only provides approximated results for the mean delay of both types of packets.

We provide a full analysis of the delay of both 1- and 2-packets in a discrete-time queue operating under the Reservation discipline when the arrival flows are assumed to be independent processes. The analysis is based on a Markovian description of the system state at the beginning of each consecutive slot and is carried out in the z-transform domain, using probability generating functions. We obtain the distribution of the delay experienced by the packets of both types during equilibrium, as well as closed-form expressions for the mean, variance and tail behaviour of those distributions.

4.1.1 Model description and system equations

To be able to study how well the reservation mechanism answers to the objective of differentiating the experienced delay between both types of packets, we now propose a concise mathematical model of a queue with the described Reservation discipline.

Let us consider a discrete-time single-server queue with infinite buffer capacity. We assume that time is divided in fixed-length intervals called *slots*, whereby one slot is exactly the service time of a packet. Following [196], there are two types of packets arriving into the queue. The packets of type 1 are delay-sensitive and will use the reserved space in the queue, while the packets of type 2 represent the best-effort traffic. Let the random variable $a_{i,k}$ denote the number of packets of type i ($i=1,2$) that arrive in the queue during slot k. In our model, we assume that the numbers of arrivals of both types are *iid* (independent and identically distributed) from slot to slot with their joint distribution given by the probability generating function (pgf) $A(z_1, z_2)$:

$$A(z_1, z_2) \triangleq \mathrm{E}[z_1^{a_{1,k}} z_2^{a_{2,k}}]\,. \tag{234}$$

As such, we allow for the numbers of arrivals of both types during the same slot to be correlated, although the arrival process is independent (and therefore uncorrelated) from slot to slot. For convenience, we will also use the marginal distributions $A_i(z)$

of the number of arrivals per slot of type i, which are given by

$$A_1(z) \triangleq \mathrm{E}[z^{a_{1,k}}] = A(z,1)\,, \tag{235}$$

$$A_2(z) \triangleq \mathrm{E}[z^{a_{2,k}}] = A(1,z)\,. \tag{236}$$

The total number of arrivals (both of type 1 and 2) during slot k is denoted by $a_{T,k} \triangleq a_{1,k}+a_{2,k}$ and has pgf

$$A_T(z) \triangleq \mathrm{E}[z^{a_{T,k}}] = A(z,z)\,. \tag{237}$$

The mean number of arrivals per slot (arrival rate) of type i follows from the moment-generating property of pgfs as

$$\lambda_i \triangleq \mathrm{E}[a_{i,k}] = A_i'(1)\,, \tag{238}$$

while the total arrival rate is

$$\lambda_T \triangleq \mathrm{E}[a_{T,k}] = A_T'(1)\,. \tag{239}$$

In Sec. 4.1.3 we will also use the higher-order derivatives evaluated for $z=1$:

$$\lambda_i' \triangleq A_i''(1)\,, \qquad \lambda_i'' \triangleq A_i'''(1)\,, \tag{240}$$

$$\lambda_T' \triangleq A_T''(1)\,, \qquad \lambda_T'' \triangleq A_T'''(1)\,. \tag{241}$$

The operation of the system under the Reservation discipline with a single reservation is visualised in Fig. 4.2. Of all the arriving packets during a certain slot, the 1-packets are always stored in the queue *before* the 2-packets. The 1-packets seize the reserved space R and make a new reserved space R at the end of the queue. Afterwards, the arriving 2-packets are stored at the end of the queue.

Let the random variable u_k denote the total number of packets present in the system at the beginning of slot k, i.e. including all 1- and 2-packets, but excluding the reserved space R. For the delay analysis in Sec. 4.1.3, we are also required to know the position of R at slot boundaries. Hence, we introduce the variable m_k, being the position of the reserved space at the beginning of slot k. For example, if Fig. 4.1 were to represent the system at the beginning of slot k, we would have $u_k = 6$ and $m_k = 3$. Since R can never enter the server at position 0 and there can be no empty places ahead of R except when the system is empty, we have

$$\begin{aligned} 1 \leqslant m_k \leqslant u_k \quad & \text{if } u_k > 0\,, \\ m_k = 1 \quad & \text{if } u_k = 0\,. \end{aligned} \tag{242}$$

An empty system is e.g. shown at the beginning of slot $k-1$ in Fig. 4.2. The basic equations that govern the evolution of these variables from one slot to the next are easily derived as follows. For the system content, we can use the well-known Lindley equation

$$u_{k+1} = (u_k - 1)^+ + a_{T,k}\,, \tag{243}$$

where $(\cdot)^+$ is the operator $\max(\cdot,0)$. From (243), it can already be seen that the total system content is the same as in the simple GI-1-1 FIFO system described in e.g. [43]

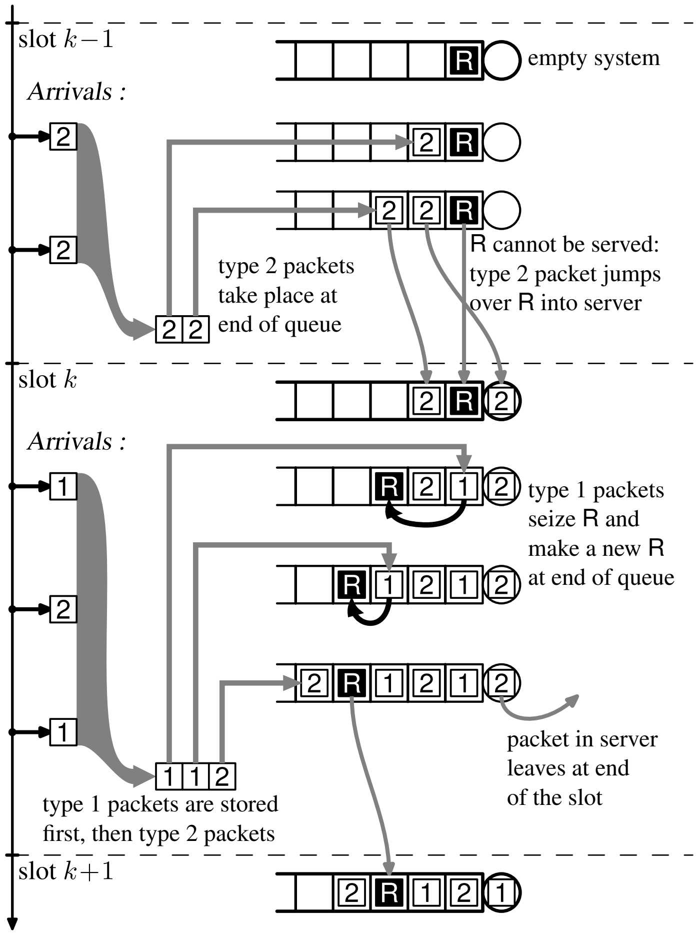


Figure 4.2: Evolution of the queue with place reservation.

with *iid* arrivals and common pgf $A_T(z)$ for the number of arrivals per slot. This is not surprising since the system content is not influenced by the order in which the packets are served. Concerning the evolution of the position of R, we distinguish two cases. First, if there is no arrival of type 1 ($a_{1,k}=0$) then the reservation at m_k persists and shifts one position to the right at the end of slot k together with all other packets in the queue (if $m_k>1$) because the packet in the server leaves. However, if $m_k=1$ and the reservation is not seized, then R stays at position 1 in the next slot. Secondly, if a batch of 1-packets ($a_{1,k}>0$) arrives during slot k, then the first packet of this batch seizes R and makes a new reservation at the end of the queue. The second 1-packet (if there is any) of the batch seizes this new reservation and makes a new one *directly* behind itself. The same happens with every remaining 1-packet in the batch and each time the position of R is increased with one place. We find:

$$m_{k+1} = \begin{cases} (m_k - 2)^+ + 1 & \text{if } a_{1,k} = 0\,, \\ (u_k - 1)^+ + a_{1,k} & \text{if } a_{1,k} > 0\,. \end{cases} \tag{244}$$

It is seen that the variables $\{m_k, u_k\}$ form an adequate Markovian description of the system state at the beginning of slot k. Therefore, we shall henceforth call them the *system state variables*.

4.1.2 Equilibrium analysis

We define the joint pgf of the system state $\{m_k, u_k\}$ at the beginning of slot k as

$$P_k(y, z) \triangleq \mathrm{E}[y^{m_k - 1}\, z^{u_k}]\,. \tag{245}$$

By means of the system equations (243) and (244), it is possible to relate the distribution of the system states at slot k and $k+1$ as follows. Using the fact that $(a_{1,k}, a_{2,k})$ are statistically independent from (m_k, u_k), we find after some standard z-transform manipulations

$$\begin{aligned} P_{k+1}(y,z) &= \mathrm{E}[y^{m_{k+1}-1} z^{u_{k+1}}] \\ &= A_1(0)\,\mathrm{E}[y^{m_{k+1}-1} z^{u_{k+1}} | a_{1,k}{=}0] + (1{-}A_1(0))\,\mathrm{E}[y^{m_{k+1}-1} z^{u_{k+1}} | a_{1,k} > 0] \\ &= \ldots \\ &= \frac{A(0,z)}{yz}\Big[(z{-}1)yP_k(0,0) + (y{-}1)P_k(0,z) + P_k(y,z)\Big] \\ &\quad + \frac{A(yz,z){-}A(0,z)}{y^2 z}\Big[(yz{-}1)P_k(1,0) + P_k(1,yz)\Big]\,, \end{aligned} \tag{246}$$

where $\mathrm{E}[X|Y]$ is the expected value of X given that Y holds. Note that because of (242) and (245) the following expressions for the probability $p_{0,k}$ that the system is empty at the beginning of slot k are equivalent:

$$p_{0,k} \triangleq \mathrm{Prob}[u_k{=}0, m_k{=}1] \;=\; \mathrm{Prob}[u_k{=}0] \;=\; P_k(0,0) \;=\; P_k(1,0)\,. \tag{247}$$

We assume that for $k \to \infty$, the system reaches equilibrium, such that the functions $P_{k+1}(y, z)$ and $P_k(y, z)$ converge to the same limiting function which we indicate by dropping the index k. Also, let (m,u), a_i and p_0 denote the system state variables, the number of type i arrivals and the probability of an empty system for an arbitrary slot during equilibrium. The condition for the system to reach such an equilibrium is the same as for the aforementioned GI-1-1 system, namely

$$\lambda_T \;<\; 1\,, \tag{248}$$

saying that the mean total arrival rate must be smaller than the maximum number of packets that can be served per slot. Taking the limit $k \to \infty$ in (246) and isolating the joint pgf $P(y, z)$ of m and u yields

$$P(y,z) \;=\; \frac{A(0,z)}{yz{-}A(0,z)}\Big[(z{-}1)yP(0,0) + (y{-}1)P(0,z)\Big]$$

$$+\frac{A(yz,z)-A(0,z)}{yz-A(0,z)}\Big[\frac{yz-1}{y}P(0,0)+\frac{1}{y}P(1,yz)\Big]. \tag{249}$$

Now, let us call $U(z)$ the marginal pgf of the system content u during equilibrium:

$$U(z) \triangleq \mathrm{E}[z^u] = P(1,z). \tag{250}$$

Because of (247) it is then clear that $U(0)=P(0,0)=p_0$. Taking the limit $y\to 1$ in (249) and isolating $U(z)$ gives

$$U(z) = p_0\,\frac{A_T(z)(z-1)}{z-A_T(z)}, \tag{251}$$

which is indeed the pgf of the system content of the GI-1-1 system [43, 44], as expected. From the normalisation condition $U(1)=1$ applied to (251), we find that the probability p_0 of having an empty system is

$$p_0 = 1-\lambda_T. \tag{252}$$

The only remaining unknown in (249) is the function $P(0,z)$ which can be determined as follows. Since $P(y,z)$ as given in (249) is a probability generating function, it must be bounded for arguments lying on the unit disc, i.e. for $|y|\leqslant 1$ and $|z|\leqslant 1$. For the same reason, it must also be analytic *inside* the unit disc (i.e. for $|y|<1$ and $|z|<1$). Therefore, $P(y,z)$ should not have singularities inside the unit disc. In theorem 2 of Sec. 4.2 we show that there is always a nonempty subset $\aleph$ of the open unit disc such that

$$z\in\aleph \;\Rightarrow\; \Big|\frac{A(0,z)}{z}\Big|<1. \tag{253}$$

If we were to choose for z^* a value lying in $\aleph$ and $y^* = A(0,z^*)/z^*$, then the denominator $yz-A(0,z)$ in (249) vanishes for $y=y^*$ and $z=z^*$. Thus, we seem to have found a singularity (y^*,z^*) of the function $P(y,z)$ lying inside the unit disc, which is impossible! The only way out of this contradiction is for the numerator in (249) to vanish for (y^*,z^*) as well. Hence, an additional relation can be found by the substitution $y=A(0,z)/z$ in the numerator of (249). After some manipulations this yields:

$$P(0,z) = \frac{p_0}{z-A(0,z)}\Big[(z-1)A(0,z) + z^2\,\frac{1-A(0,z)}{A(0,z)}\,\frac{A(0,z)-A(A(0,z),z)}{A(0,z)-A(A(0,z),A(0,z))}\Big]. \tag{254}$$

It will be the main subject of the next section to prove that this expression is valid not only for $z\in\aleph$, but for *every* z in the unit disc, whenever the function $A(z_1,z_2)$ satisfies some broad regularity conditions. Finally, after eliminating all occurrences of the function $P(\cdot)$ in the right hand side of (249) using (251) and (254), we find the following explicit expression for the system state distribution during an arbitrary slot in equilibrium:

$$P(y,z) = \frac{p_0}{yz-A(0,z)}\Big[A(0,z)(z-1)\frac{yz-A(0,z)}{z-A(0,z)} + z(yz-1)\frac{A(yz,z)-A(0,z)}{yz-A_T(yz)}$$

$$+ z^2(y-1)\frac{1 - A(0,z)}{z - A(0,z)}\,\frac{A(0,z) - A(A(0,z),z)}{A(0,z) - A_T(A(0,z))}\Bigg]\,. \quad (255)$$

4.1.3 Distribution of the packet delay

Let us now consider the arrivals of one type only, say type i. Of all the packets of type i arriving to the system, we choose an arbitrary packet and tag it as packet $\mathcal{P}$. Also, let us mark the arrival slot of $\mathcal{P}$ as slot I. We define the delay d_i as the number of full slots between the end of the slot in which $\mathcal{P}$ arrives (slot I) and the end of the slot in which $\mathcal{P}$ departs from the queue.

We can quantify the delay d_i as the amount of slots required to process the unfinished work present in the queue directly *after* $\mathcal{P}$ is stored in the queue, keeping in mind that the packets are stored one by one, first the 1-packets, then the 2-packets. Clearly, this unfinished work foremost depends on the system state (m_I, u_I) at the beginning of slot I. Hence, we first need an expression for the system state distribution $P_I(y,z)$ in slot I. Now, it has been argued before (see e.g. [43]) that due to the uncorrelated (*iid*) nature of the packet arrival process, the system state as 'seen' by an *arbitrary arriving packet* of either type has the same distribution as the system state in an *arbitrary slot*. Therefore, we can immediately use the result (255) of the previous Section for the joint pgf $P(y,z)$ as it is also the distribution of the system state during the arrival slot of an arbitrary packet,

$$(m_I, u_I) \overset{d}{=} (m,u)\,, \qquad P_I(y,z) = P(y,z)\,. \quad (256)$$

Based on this observation, we can derive the pgfs $D_1(z)$ and $D_2(z)$ of the packet delay for the specific cases in which $\mathcal{P}$ is of type 1 and of type 2 respectively. Because of (256), we may write m and u instead of m_I and u_I for the system state variables as seen by an arriving batch containing $\mathcal{P}$.

Before we can proceed, we must introduce some further notation. First, let a_1^I and a_2^I be the total numbers of arrivals of type 1 and 2 respectively during the arrival slot I of $\mathcal{P}$. Note that these variables do *not* have the same joint distribution as a_1 and a_2. If $\mathcal{P}$ is of type i, the joint mass function of the numbers of arrivals of type 1 and 2 during the arrival slot of $\mathcal{P}$ is given by (see [43]):

$$\text{Prob}[a_1^I = j_1, a_2^I = j_2] = \frac{j_i}{\lambda_i}\text{Prob}[a_1 = j_1, a_2 = j_2] \qquad , i = 1,2\,. \quad (257)$$

Next, again with i indicating the type of packet $\mathcal{P}$, let ℓ_i be the number of packets of type i that arrive during slot I and are stored in the queue no later than (but including) $\mathcal{P}$, as shown in Figs. 4.3 and 4.4. The pgf $L_i(z)$ of ℓ_i is found as (see [43])

$$L_i(z) = \frac{z(1 - A_i(z))}{\lambda_i(1-z)} \qquad , i = 1,2\,, \quad (258)$$

by considering the fact that the pgf of the number of type i arrivals a_i^I in slot I is *not* $A_i(z)$, but rather $zA_i'(z)/\lambda_i$ and that $\mathcal{P}$ could be *any* of the a_i^I arrivals with equal probability.

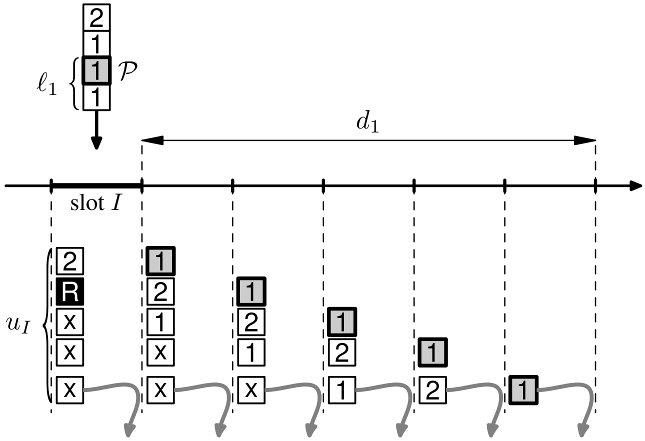


Figure 4.3: Delay of a tagged packet $\mathcal{P}$ of type 1 arriving in slot I. Only the packets scheduled ahead of $\mathcal{P}$ are shown.

Delay of type 1 packets

Let us now consider the case in which $\mathcal{P}$ is an arbitrary packet of type 1. As shown in Fig. 4.3, the delay d_1 of $\mathcal{P}$ clearly depends on the system state (m,u) at the beginning of slot I as well as on the number and order of other arrivals of type 1 during slot I. Since exactly one packet is served every slot, the delay of $\mathcal{P}$ is equal to its position in the queue when stored during I. To characterise the delay d_1 we must again distinguish between two cases. First, if there are no type 1 packets to be stored during I prior to $\mathcal{P}$ (i.e. $\ell_1 = 1$) then $\mathcal{P}$ seizes the reservation at position m. Otherwise, if $\ell_1 > 1$, that reservation is already taken by the first of the $\ell_1 - 1$ previously stored 1-packets and $\mathcal{P}$ seizes the reservation made by the $(\ell_1 - 1)$th 1-packet. Hence,

$$d_1 = \begin{cases} m & \text{if } \ell_1 = 1\,, \\ (u-1)^+ + \ell_1 & \text{if } \ell_1 > 1\,. \end{cases} \tag{259}$$

From (258) we find

$$\text{Prob}[\ell_1 = 1] = \left.\frac{L_1(z)}{z}\right|_{z=0} = \frac{1 - A_1(0)}{\lambda_1}\,. \tag{260}$$

Observing that ℓ_1 is independent of m and u, we can thus calculate the pgf $D_1(z)$ as

$$\begin{aligned} D_1(z) &\triangleq \text{E}[z^{d_1}] \\ &= \text{E}[z^m | \ell_1 = 1]\,\text{Prob}[\ell_1 = 1] + \text{E}[z^{(u-1)^+ + \ell_1} | \ell_1 > 1]\,\text{Prob}[\ell_1 > 1] \\ &= \ldots \\ &= \frac{1 - A_1(0)}{\lambda_1}\, zP(z,1) + \frac{p_0}{\lambda_1}\, z\, \frac{A_1(z) - z - (1-z)A_1(0)}{z - A_T(z)}\,, \end{aligned} \tag{261}$$

where we have used (251). The function $P(z,1)$ is simply the marginal pgf of the reservation position m and can easily be found from (255) by subsequent substitutions $z \to 1$ and $y \to z$:

$$P(z,1) = \mathrm{E}[z^m] = \frac{(z-1)p_0}{z-A_1(0)} \left[\frac{A_1(z)-A_1(0)}{z-A_T(z)} + \phi \right], \tag{262}$$

where we have defined the constant ϕ as

$$\phi \triangleq \frac{A_1(0) - A_1(A_1(0))}{A_1(0) - A_T(A_1(0))}. \tag{263}$$

Finally, (261) yields

$$\begin{aligned} D_1(z) = \frac{p_0}{\lambda_1} z \Bigg[& \frac{A_1(z) - z + (z-1)A_1(0)}{z - A_T(z)} \\ & + \frac{(z-1)(1-A_1(0))}{z - A_1(0)} \left(\frac{A_1(z)-A_1(0)}{z - A_T(z)} + \phi \right) \Bigg]. \end{aligned} \tag{264}$$

Delay of type 2 packets

If the tagged packet $\mathcal{P}$ is of type 2, the calculation of the delay is slightly more complicated, as shown in Fig. 4.4. Indeed, unlike the previous case, the delay of a type 2 packet is not entirely independent of the arrivals *after* slot I. An important issue to consider is that immediately after slot I, the reservation R is always positioned *ahead* of $\mathcal{P}$. The distribution of d_2 depends on whether that reserved space is seized by one of the future arriving 1-packets *before* $\mathcal{P}$ reaches the server, or not. If so, the delay is increased by one slot to account for the additional delay introduced by that 1-packet that jumped over $\mathcal{P}$. Specifically, let γ_n be a Bernoulli random variable that is 1 if one or more arrivals of type 1 occur during n subsequent slots and is 0 otherwise. Evidently, because of the independent nature of the arrivals, the distribution of γ_n is

$$\begin{aligned} \mathrm{Prob}[\gamma_n = 0] &= \big(A_1(0)\big)^n, \\ \mathrm{Prob}[\gamma_n = 1] &= 1 - \big(A_1(0)\big)^n. \end{aligned} \tag{265}$$

Let us first assume that the reserved position is never seized after slot I. The delay of $\mathcal{P}$ would then be given by

$$v \triangleq (u-1)^+ + a_1^I + \ell_2. \tag{266}$$

To obtain the pgf $V(z)$ of v, we require the distribution of $a_1^I + \ell_2$. Based on (257) and the fact that $\mathcal{P}$ could be *any* of the a_2^I arrivals of type 2 during slot I with equal probability, we find for the joint pgf $F(x,y)$ of a_1^I and ℓ_2:

$$F(x,y) \triangleq \mathrm{E}[\,x^{a_1^I}\, y^{\ell_2}\,] = \frac{y}{1-y}\, \frac{A_1(x)-A(x,y)}{\lambda_2}, \tag{267}$$

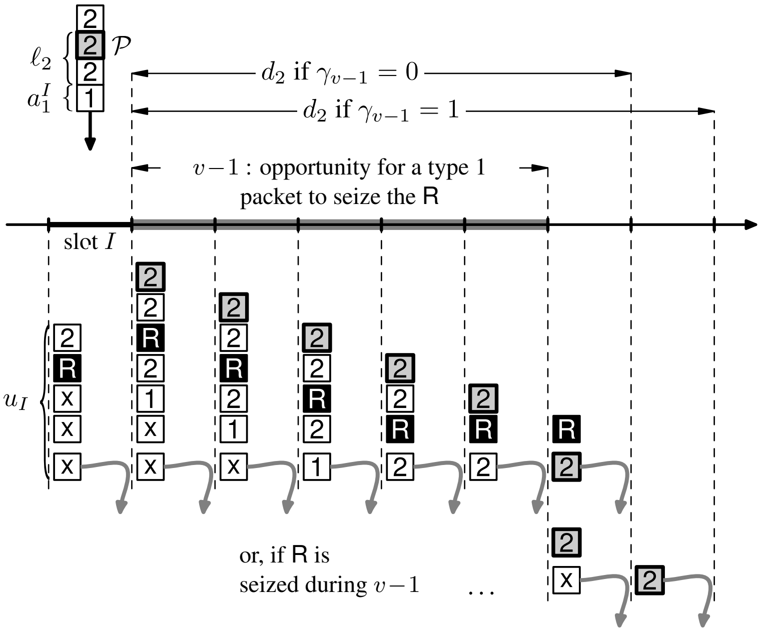


Figure 4.4: Delay of a tagged packet $\mathcal{P}$ of type 2 arriving in slot I. Only the packets scheduled ahead of $\mathcal{P}$ are shown. The delay is increased by one slot if the R is seized before $\mathcal{P}$ reaches the server.

and the pgf of $a_1^I + \ell_2$ is then simply given by $F(z,z)$. From (266), (267) and (251) we can easily obtain $V(z)$ as

$$V(z) \;=\; \mathrm{E}[z^{(u-1)^+}]\, F(z,z) \;=\; \frac{p_0}{\lambda_2}\, z\, \frac{A_T(z) - A_1(z)}{z - A_T(z)}\,, \tag{268}$$

since u is independent of a_1^I and ℓ_2. However, as explained above, the delay may be increased from v to $v{+}1$ if there is an arrival of type 1 before $\mathcal{P}$ enters the server, i.e. during the $v{-}1$ slots after slot I. Therefore, we can write

$$d_2 \;=\; v + \gamma_{v-1}\,. \tag{269}$$

Using (265) we can thus write the pgf of the delay for an arbitrary packet of type 2 as

$$D_2(z) \;=\; zV(z) + \frac{1-z}{A_1(0)} V(zA_1(0))\,. \tag{270}$$

Note that a good heavy-traffic approximation for $D_2(z)$ is $zV(z)$ since in that case the reserved packet is very likely to be seized during the period $v-1$ and hence $\mathrm{Prob}[\gamma_{v-1}=1] \approx 1$. Even under low traffic conditions, this approximation is a tight upper bound for the type 2 delay. Finally, substitution of (268) into (270) yields

$$D_2(z) \;=\; \frac{p_0}{\lambda_2}\, z \left[z\frac{A_T(z) - A_1(z)}{z - A_T(z)} + (1{-}z)\frac{A_T(zA_1(0)) - A_1(zA_1(0))}{zA_1(0) - A_T(zA_1(0))} \right]. \tag{271}$$

Moments and Tail of the Packet Delay

Once we have obtained the pgfs of the packet delay for both types, we can derive several interesting performance measures from them that are useful in practice. First of all, invoking the moment generating property on the obtained pgfs (264) and (271) yields closed-form expressions for the moments of the packet delay up to any desired order, although the subsequent differentiations quickly become very involved for higher-order moments. For instance, the mean packet delays $\mathrm{E}[d_i] = D_i'(1)$ follow as

$$D_1'(1) \;=\; 2 - \frac{1-p_0\,\phi}{\lambda_1} + \frac{\lambda_T'}{2p_0} + \frac{\lambda_1'}{2\lambda_1}\,, \tag{272}$$

and

$$D_2'(1) \;=\; 2 + \frac{(1-\lambda_1)\lambda_T' - p_0\lambda_1'}{2p_0\lambda_2} - \frac{V(A_1(0))}{A_1(0)}\,, \tag{273}$$

where the function $V(z)$ is given by (268) and ϕ by (263). The second-order moments $\mathrm{Var}[d_i] = D_i''(1) + D_i'(1) - D_i'^{\,2}(1)$ are found by differentiating (264) and (271) a second time and taking the limit $z \rightarrow 1$, which yields

$$\begin{aligned} D_1''(1) \;=\; & \frac{\lambda_T'' + 3\lambda_T'}{3p_0} + \frac{\lambda_1'' + 3\lambda_1'}{3\lambda_1} + \frac{\lambda_T'^{\,2}}{2p_0^2} + \frac{2(\lambda_1-1) + 2p_0\phi + \lambda_1'}{\lambda_1} \\ & + \lambda_T' \frac{2(\lambda_1-1) + \lambda_1'}{2p_0\lambda_1} + 2\frac{1-\lambda_1 - p_0\phi}{\lambda_1(1-A_1(0))}\,, \end{aligned} \tag{274}$$

and

$$\begin{aligned} D_2''(1) \;=\; & 2 + \frac{\lambda_T'' - \lambda_1''}{3\lambda_2} + \frac{\lambda_T''}{3p_0} + \frac{\lambda_T'\big(\lambda_T' - \lambda_1' + 2(1-\lambda_1)\big)}{2p_0\lambda_2} \\ & + \frac{\lambda_T'^{\,2}}{2p_0^2} + \frac{\lambda_T' - 2\lambda_1'}{\lambda_2} + \frac{\lambda_T'}{p_0} - 2V'(A_1(0))\,. \end{aligned} \tag{275}$$

Note that to obtain (272)–(275), multiple applications of de l'Hôpital's rule were required when taking the limit $z \rightarrow 1$.

To derive the tail distribution of the packet delay, we use an approximation technique which is known to yield very accurate results [43,46]. Specifically, from the inversion formula for z-transforms follows that the probability mass function $\mathrm{Prob}[d_i = n]$ $(i = 1, 2)$ can be expressed as a weighted sum of negative nth powers of the poles of $D_i(z)$. Since all these poles have a modulus larger than 1, $\mathrm{Prob}[d_i = n]$ is dominated by the contribution of the pole $z_{d,i}$ with the smallest modulus. It is shown [46] that this 'dominant' pole $z_{d,i}$ must necessarily be real and positive in order to ensure a nonnegative probability mass function. As such, the probability for a packet of type i to experience a delay of n slots can be expressed by the following geometric form for sufficiently large values of n:

$$\mathrm{Prob}[d_i = n] \;\cong\; -\theta_i z_{d,i}^{-n-1}\,, \tag{276}$$

where θ_i is the residue of $D_i(z)$ in the point $z = z_{d,i}$, i.e.

$$\theta_i \;=\; \mathrm{Res}_{z_{d,i}} D_i(z) \;=\; \lim_{z \rightarrow z_{d,i}} (z - z_{d,i}) D_i(z)\,. \tag{277}$$

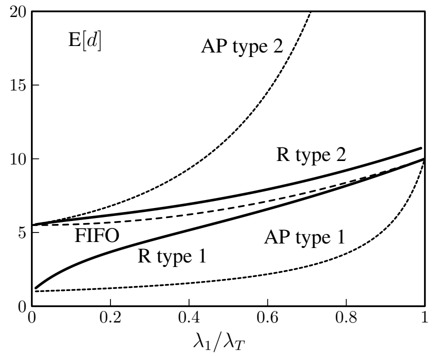


Figure 4.5: Mean delay for both types of packets as a function of λ_1/λ_T in case of the Reservation discipline, Absolute Priority and FIFO.

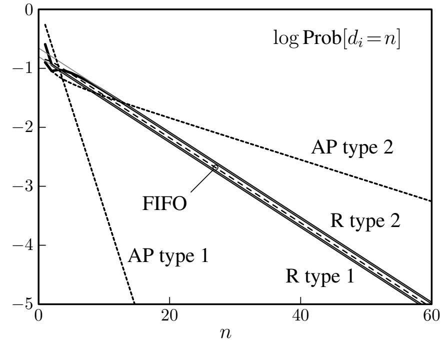


Figure 4.6: $\log \text{Prob}[d_i = n]$ of the delay for both types of packets in case of the Reservation discipline (R), Absolute Priority (AP) and FIFO.

To identify $z_{d,i}$ and θ_i $(i=1,2)$, we can proceed as follows. Both in case of (264) and (271), it can be shown that the dominant pole is always given by the smallest zero larger than 1 of the factors $z-A_T(z)$ in their respective denominators. Hence, $D_1(z)$ and $D_2(z)$ have the *same* dominant pole which we denote by

$$z_d \triangleq z_{d,1} = z_{d,2}\,. \tag{278}$$

We call z_d^{-1} the *decay rate* of the delay distributions, because of the geometric form in (276). Clearly, z_d satisfies

$$z_d - A_T(z_d) = 0\,, \tag{279}$$

and can be determined numerically by using e.g. the Newton-Raphson method. The residues θ_i of $D_i(z)$ $(i = 1,2)$ in the point $z = z_d$ are found from (277). Using de l'Hôpital's rule, we find

$$\begin{aligned}\theta_1 = {} & \frac{p_0\, z_d}{\lambda_1(1-A_T'(z_d))(z_d-A_1(0))}\Big[A_1(0)(z_d-1)^2 \\ & + A_1(z_d)(1-A_1(0))(z_d-1) + (z_d-A_1(0))(A_1(z_d)-z_d)\Big]\,,\end{aligned} \tag{280}$$

and

$$\theta_2 = \frac{p_0}{\lambda_2}\, z_d^2\, \frac{A_T(z_d) - A_1(z_d)}{1 - A_T'(z_d)}\,, \tag{281}$$

respectively.

4.1.4 A numerical example

In order to illustrate the impact of the Reservation discipline on the equilibrium distribution of the packet delays d_1 and d_2, we now consider a short practical example. Both

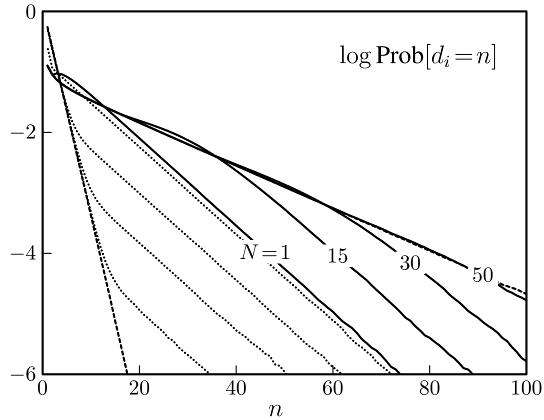


Figure 4.7: Simulated mass function of the delay for type 1 packets (dotted lines) and type 2 packets (solid lines) for increasing number of reservations N in the queue. The dashed lines indicate the delays for AP.

Figs. 4.5 and 4.6 are plots of the delays in case the joint distribution of the arrivals has the following form:

$$A(z_1, z_2) = \frac{1}{1+\lambda_1 - z_1\,\lambda_1} \cdot e^{\lambda_2(z_2 - 1)}\,, \tag{282}$$

i.e. the numbers of arrivals per slot of type 1 and 2 are independent and have a geometric and Poisson distribution respectively, with partial loads λ_1 and λ_2. In Fig. 4.5, we have chosen the total load $\lambda_T = \lambda_1 + \lambda_2$ equal to 0.9 and plotted the mean delays $\mathrm{E}[d_1]$ and $\mathrm{E}[d_2]$ for 1- and 2-packets as functions of the traffic mix λ_1/λ_T. The obtained values for FIFO and both types of packets under AP (see [196]) are also shown. Note that in case of FIFO, we considered the delay of an arbitrary packet *regardless* of its type. Therefore there is only one curve for FIFO instead of two for the Reservation discipline and AP, where we considered the delay of each type separately. As expected, under the Reservation discipline, the type 1 delay is reduced in comparison with FIFO but not as much as would have been obtained with AP. Also, the delay of the best-effort traffic (type 2) does not increase as drastically as with AP.

In Fig. 4.6, we show the mass functions Prob$[d_i = n]$ on a logarithmic scale for the case $\lambda_1/\lambda_T = 0.5$, i.e. for both flows having the same partial load: $\lambda_1 = \lambda_2 = 0.45$. These plots were obtained by numerical inversion of the pgfs (264) and (271) as discussed in [20]. The thin gray lines indicate the geometric approximations given by (276), which are clearly very accurate. Again, the results are compared to that of FIFO and AP. We see that the delays of type 1 and 2 have a geometric tail with the same decay rate (i.e. both curves have the same slope) as predicted by (278). Also, we see that the decay rate is the same as that of FIFO. Note that in the case of AP, it is known that the tail of the type 2 delay may have a non-geometrical decay (see [196]).

One may conclude from these numerical examples that the delay differentiation between the two types of packets obtained with the described reservation mechanism is rather limited. An obvious extension of the model in this section is to introduce multiple reservation places in the queue. The reservation discipline with N reservation places in the queue is treated in detail in Section 4.3. Fig. 4.7 gives the results of a preliminary simulation experiment, that showed us what to expect if the number of reservations is increased. In short, the operation of the multi-reservation queue is as follows. When the queue is empty, the reservations are placed at positions 1 to N at the front of the queue. A type 1 arrival will now seize the reservation with the lowest position number and make a new reservation at the end of the queue as before. With more reserved spaces, a 1-packet can skip more already stored 2-packets when it is placed in the queue. Hence, the delay differentiation between the two types of packets increases with the number of reserved spaces. For very large N, the delay behaviour tends to that of the AP discipline as is shown in Fig. 4.7 where the logarithmic delay mass functions for $N = 1, 15, 30, 50$ is plotted. We see that the amount of reserved spaces N is an appropriate parameter by which the delay differentiation can be carefully controlled between the two extreme cases of FIFO and AP.

4.2 Some remarks on the determination of $P(0, z)$

The attentive reader may raise some questions about the way in which we have derived the unknown function $P(0, z)$ on page 149. In our argumentation there, we arrive at expression (254) for $P(0, z)$ starting from the assumption that a nonempty region $\aleph$ defined by (253) exists in the open unit disc. We already mentioned there that 'it is possible' to prove the existence of such a region $\aleph$ but we did not go into detail at the time. It turns out that this region can be chosen to have the shape of a ring as depicted in Fig. 4.8. The actual proof is the subject of the forthcoming theorem 2.

Secondly, once we agree that $\aleph$ exists, it was convincingly shown that $P(0, z)$ must be given by (254), at least if $z \in \aleph$. But what if $z \notin \aleph$? Is the right-hand side of (254), rewritten here as

$$f(z) = \frac{p_0}{z - A(0, z)} \Big[(z-1)A(0, z) + z^2 \frac{A(0, z) - A(A(0, z), z)}{A(0, z)} \frac{1 - A(0, z)}{A(0, z) - A_T(A(0, z))} \Big], \tag{283}$$

then still the correct representation of the (partial) pgf $P(0, z)$? The answer is yes, which we can prove by means of theorem 3 and the argument of analytic continuation. Indeed, we know that for $z \in \aleph$, $P(0, z)$ is equal to $f(z)$. Hence, $f(z)$ is analytic in $\aleph$ precisely *because* it is the representation of a partial pgf there (any pgf or partial pgf is analytic inside the unit disc). Theorem 3 establishes that, under some regularity conditions for the pgf $A(z_1, z_2)$, $f(z)$ is analytic not only in $\aleph$ but in a region D_ϵ slightly larger than the unit disc, see Fig. 4.9. Therefore, an analytic representation of $P(0, z)$ in D_ϵ can only be given by $f(z)$.

4.2.1 Assumptions

Before we proceed, we introduce some additional notations and assumptions besides those of Sec. 4.1.1. Let $D = \{z : |z| \leqslant 1\}$ be the unit disc, $\mathring{D} = \{z : |z| < 1\}$

the open interior of the unit disc and $\partial D = \{z : |z| = 1\}$ the unit circle. Also, let $D_\epsilon = \{z : |z| < 1+\epsilon\}$ for some $\epsilon > 0$. By a_T we denote the total number of arrivals (both of type 1 and 2) with pgf $A_T(z)$ and mass function

$$a_T(h) \triangleq \text{Prob}[a_T = h]\,, \qquad h = 0, 1, \ldots\,.$$

The joint mass function of the number of arrivals of both types is denoted by

$$a(h_1, h_2) \triangleq \text{Prob}[a_1 = h_1, a_2 = h_2]\,,$$

such that

$$A(z_1, z_2) = \sum_{h_1=0}^{+\infty} \sum_{h_2=0}^{+\infty} a(h_1, h_2)\, z_1^{h_1}\, z_2^{h_2}\,.$$

Note that we have dropped the slot index k for convenience. We make the following assumptions regarding the distribution $A(z_1, z_2)$ of the number of 1- and 2-arrivals per slot, in order to exclude some trivial or pathological cases:

❶ Arrivals of either type occur with non-zero probability, i.e.

$$\begin{aligned} \text{Prob}[a_1 = 0] &= A(0,1) = A_1(0) < 1\,, \\ \text{Prob}[a_2 = 0] &= A(1,0) = A_2(0) < 1\,. \end{aligned} \tag{284}$$

As such, we exclude the cases in which either $\lambda_1 = 0$ or $\lambda_2 = 0$, i.e. in which there are no 1- or 2-arrivals respectively. Obviously, if there are only arrivals of one type, their delay distribution follows from the analysis of the GI-D-1 system.

❷ We also assume that the system is stable, i.e. seen over sufficiently large time periods, the number of arriving packets does not exceed the number of possible departures. A *sufficient* condition for stability is

$$\lambda_T = A_T'(1) < 1\,, \tag{285}$$

which implies among other things that

$$a(0,0) = A(0,0) = A_T(0) = a_T(0) > 0\,.$$

Note however that the requirement $\lambda_T < 1$ is not a *necessary* condition for stability. Consider the trivial case in which there is exactly one packet arrival per slot, i.e. $a_T = 1$. In that special case, the system is stable although $\lambda_T = 1$ and $a(0,0) = 0$. Then, there is always exactly one packet in the queue at the beginning of each slot and every packet has a delay of 1 slot.

❸ The pgf $A(z_1, z_2)$ is analytic not only in $\mathring{D}$ but also on ∂D, or more accurately:

$$\exists \epsilon^* > 0 : A(z_1, z_2) \text{ is analytic for } z_1, z_2 \in D_{\epsilon^*} = \{z : |z| < 1+\epsilon^*\}\,. \tag{286}$$

This is a strong regularity condition. There exist some exceptional cases of pgfs $A(z_1, z_2)$ which are bounded but not analytic on ∂D. A simple example of such a pgf in only one variable is

$$f(z) = 1 - \sqrt{\frac{1-z}{2}}\,,$$

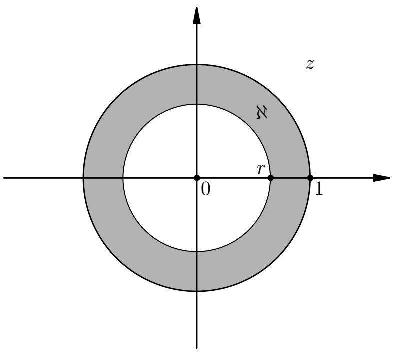


Figure 4.8: The region ℵ within the unit disc D of the complex z-plane.

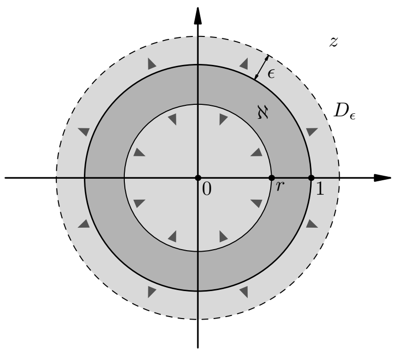


Figure 4.9: $P(0, z)$ is given by the unique analytic continuation of $f(z)$ from ℵ to D_ϵ.

which has a branch point in $z=1$. To see that $f(z)$ is indeed a proper pgf observe that $f(1)=1$ and that its coefficients $f_k \triangleq [z^k]f(z)$

$$f_k = \sqrt{2}\frac{1 \cdot 3 \cdot 5 \cdots (2k-3)}{k!}2^{k-1},$$

are all nonnegative. Notice also that the condition (286) assures that all moments of the arrival distribution exist.

In aid of the exposition in this section, we also recall the following known facts from the theory of complex variables [36, 86, 99].

Fact 1. *If f_1 and f_2 are analytic in Ω then $f_1(z) \cdot f_2(z)$ as well as every linear combination of f_1 and f_2 are analytic in Ω.*

Fact 2. *If f is analytic in Ω then $1/f$ is analytic in $\Omega \setminus \{z : f(z)=0\}$.*

Fact 3. *If f_1 is analytic in Ω and f_2 analytic in Ω' with $\Omega' \supset \{f_1(z) : z \in \Omega\}$, then $f_2(f_1(z))$ is analytic in Ω.*

Fact 4 (Rouché's theorem)**.** *If f_1 and f_2 are analytic in Ω and C is a closed contour in Ω with $|f_2(z)| < |f_1(z)|$ on C, then f_1 and f_1+f_2 have the same number of roots inside C.*

Fact 5 (Analytic continuation)**.** *If f_1 and f_2 are analytic in the open connected regions Ω_1 and Ω_2 respectively, and $f_1 = f_2$ on a non-empty open subset of $\Omega_1 \cap \Omega_2$ that is also connected, then f_2 is called an analytic continuation of f_1 to Ω_2 and vice versa. If it exists, the analytic continuation of f_1 to Ω_2 is unique. In other words, there exists a unique function F analytic in $\Omega_1 \cup \Omega_2$ that coincides with f_1 in Ω_1 and with f_2 in Ω_2.*

4.2.2 The nonempty set ℵ always exists

Theorem 2. *Suppose $\aleph = \{z : r < |z|, |z| < 1\}$ with*

$$r = \frac{A(0,0)}{A(0,0) + 1 - A_1(0)}, \tag{287}$$

then: (i) $r<1$*, i.e.* $\aleph$ *is an open non-empty annulus,*

(ii) In $\aleph$*, we have the bound*

$$\forall z \in \aleph : \left|\frac{A(0,z)}{z}\right| < 1\,. \tag{288}$$

Proof.

(i) Because of (284), we have $1-A_1(0) > 0$ in the denominator of (287). Hence $r<1$.

(ii) For all z in D we have

$$|z| \leqslant 1 \Rightarrow \begin{cases} |z|^n = 1 & \text{if } n=0\,, \\ |z|^n \leqslant |z| & \text{if } n>0\,. \end{cases}$$

Therefore,

$$\begin{aligned} |A(0,z)| &= \left|\sum_{n=0}^{+\infty} a(0,n) z^n\right| \\ &\leqslant \sum_{n=0}^{+\infty} a(0,n)|z|^n = a(0,0) + \sum_{n=1}^{+\infty} a(0,n)|z|^n \\ &\leqslant A(0,0) + \sum_{n=1}^{+\infty} a(0,n)|z| \\ &= A(0,0) + |z|\big(A_1(0) - A(0,0)\big)\,, \end{aligned} \tag{289}$$

and

$$\left|\frac{A(0,z)}{z}\right| \leqslant \frac{A(0,0)}{|z|} + \big(A_1(0) - A(0,0)\big)\,.$$

This upper bound increases towards infinity if $|z| \to 0$. However, there is a non-empty subset of D in which this bound is smaller than one:

$$\frac{A(0,0)}{|z|} + \big(A_1(0) - A(0,0)\big) < 1 \quad \Leftrightarrow \quad |z| > \frac{A(0,0)}{A(0,0)+1-A_1(0)} = r\,.$$

Thus, at least for all $r < |z| < 1$ (i.e. for $z \in \aleph$), we have $\left|\frac{A(0,z)}{z}\right| < 1$.

□

Using this theorem, we have shown on page 149 that for $z \in \aleph$, the pgf $P(0,z)$ is given by the function $f(z)$ defined in (283). So at least in $\aleph \subset \mathring{D}$, $f(z)$ must be analytic because it is a pgf there. In fact, in order to conclude this, we would have to prove that the system state constitutes a Markov chain that is *ergodic*, which means that every state can be expected to reach any other state within finite time (see [82]). Sufficient conditions for ergodicity are e.g. stated in Foster's theorem [88,158] and are satisfied if condition (285) for stability holds. Our next task is to show that the region where $f(z)$ is analytic extends beyond D, at least to some region D_ϵ, $\epsilon > 0$.

4.2.3 The analytic continuation of $f(z)$ to D_ϵ

We have seen that the pgf $P(y,z)$ of the system state features the function $P(0,z)$, see e.g. (249). So, as far as they exist, obtaining the moments of the system state requires evaluating the derivatives of $P(0,z)$ to z in the point $z=1$. However, the derivatives are only uniquely defined if the function $f(z)$ is analytic in (a neighbourhood of) $z=1$. Since $z=1$ is not in $\aleph$, we have to check this first. On the other hand, to invert $f(z)$ (i.e. obtain the coefficients $[z^n]f(z)$), we already have sufficient information:

$$[z^n]f(z) \;=\; \frac{1}{2\pi\jmath}\oint_{C_0}\frac{f(z)}{z^{n+1}}\mathrm{d}z\,.$$

where C_0 is a closed contour around $z=0$ lying in a region where $f(z)$ is analytic. Choosing $C_0 \in \aleph$ is always possible, and allows to compute the coefficients. But if we want to invert $f(z)$ by using the residue theorem, we need to show that $f(z)$ is analytic in a neighbourhood of $z=0$, since we require the derivatives in that point.

$$[z^n]f(z) \;=\; \frac{1}{2\pi\jmath}\oint_{C_0}\frac{f(z)}{z^{n+1}}\mathrm{d}z \;=\; \mathrm{Res}_{z=0}\frac{f(z)}{z^{n+1}} \;=\; \frac{1}{n!}\frac{\mathrm{d}^n}{\mathrm{d}z^n}f(z)\Big|_{z=0}\,.$$

Also, the Cauchy theorem requires that $f(z)$ is analytic within the contour C_0 except for a number of isolated points. Again, analyticity in $z=0$ has to be confirmed first, since $\aleph$ does not contain this point. In the following theorem, we rigorously show that $f(z)$ as given in (283) is analytic in a region D_ϵ ($\epsilon<\epsilon^*$), based on the assumptions we made on $A(z_1,z_2)$. This region contains a neighbourhood of both $z=1$ and $z=0$. Prior to theorem 3, we mention some lemmas first.

Lemma 1. *There exists $\epsilon'>0$ such that for all $z\in D_{\epsilon'}$, we have $|A(0,z)|<1$.*

Proof. For $z\in D$, the proof is similar to that of theorem 2. If $|z|\leqslant 1$, the bound (289) is valid which becomes

$$|A(0,z)| \leqslant A(0,0) + |z|\big(A_1(0)-A(0,0)\big) \leqslant A_1(0) < 1\,,$$

due to assumption (284). Now, since $A(0,z)$ is analytic in a disc around $z=0$ with radius at least a little bit larger than 1 (assumption (286)), it is also continuous there and all its derivatives exist and are continuous as well. Therefore, since we know that $|A(0,z)|<1$ on the unit disc, there must exist an ϵ' such that $|A(0,z)|<1$ in $D_{\epsilon'}$ too. □

Lemma 2. *There exists $\epsilon'>0$ such that the function $A(0,z)-z$ has exactly one zero in $D_{\epsilon'}$.*

Proof. Again, since $A(z_1,z_2)$ is analytic in D_{ϵ^*}, so is the function $f_1(z)=A(0,z)$. Let us call $f_2(z)=-z$ which obviously is analytic everywhere. Therefore, both f_1 and f_2 are continuous with continuous derivatives in D_ϵ. Also, from lemma 1 we know that

$$|f_1(z)| = |A(0,z)| < |z| = |f_2(z)| = 1\,,$$

on the unit circle ∂D. Because both f_1 and f_2 are continuous, the inequality must hold for at least a little bit outside ∂D, or in other words, there exists $\epsilon' > 0$ such that

$$\forall z \in \bar{D}_{\epsilon'} : |f_1(z)| = |A(0,z)| < |z| = |f_2(z)|,$$

where $\bar{D}_{\epsilon'} = \{z : |z| \leqslant 1+\epsilon'\}$. Now we can invoke Rouché's theorem for the functions f_1 and f_2 using the contour $C = \partial D_{\epsilon'}$. Hence, we can conclude that $-z$ and $A(0,z)-z$ have the same number of zeroes in $D_{\epsilon'}$. Clearly, $-z$ only has one zero ($z=0$) in $D_{\epsilon'}$, so $A(0,z)-z$ must have exactly one zero there as well. □

Lemma 3. *The function* $x - A_T(x)$ *has no zeroes in* $\mathring{D}$.

Proof. Recall that $A_T(x)$ is a pgf for which $A_T(1) = 1$. Usually, to prove that there are no roots of $x - A_T(x)$ in $\mathring{D}$, people use Rouché's theorem with contour $C = \partial D$. However, this requires that $|A_T(x)| < |x|$ everywhere on C, which obviously is not the case for $x = 1$ and possibly some other points on C (consider e.g. $A_T(x) = (1+x^2)/2$ for which we have $A_T(1) = A_T(-1) = 1$). In the appendix of [43], this problem was alleviated by extending C with a small arc $C_\epsilon = \{x : x = 1 + \epsilon e^{\jmath\alpha}, \alpha \in [-\frac{\pi}{2}, \frac{\pi}{2}]\}$ *around* $x = 1$ outwards of D and by checking that $|A_T(x)| < |x|$ everywhere on $C' = C \setminus \{z=1\} \cup C_\epsilon$. In [115] the Rouché theorem was restated in such a way that the extra work around $x = 1$ is no longer necessary but, as in [43], *still* requires that $|A_T(x)| < 1$ in all other points of ∂D, which is a rather limiting assumption. However, it is not difficult to extend the proof of [43] to the case where the pgf $A_T(x)$ becomes equal to 1 in multiple points of the unit circle ∂D.

We call the *support* of the random variable a_T with pgf $A_T(x)$ the set of values that a_T can assume with non-zero probability, i.e.

$$\text{support}(a_T) \triangleq \{h \geqslant 0 : \text{Prob}[a_T = h] > 0\}.$$

In terms of the pgf, this is the set of all powers h for which the coefficient $a_T(h)$ of z^h in the expansion of $A_T(x)$ is non-zero. Further, let M be the greatest common divisor of this set,

$$M \triangleq \gcd \text{support}(a_T),$$

then it is seen that $A_T(x)$ becomes equal to 1 on ∂D in exactly the points $x_n = e^{\jmath 2\pi n/M}$ ($n = 0, 1, \ldots, M-1$) and has a modulus smaller than 1 on the rest of ∂D. Furthermore, due to assumption (285), M must be finite. Note that since $A_T(x)$ is analytic on ∂D, if $A_T(x)$ would be equal to 1 in an *infinite* number of points of ∂D, then $A_T(x) = 1$ *everywhere*, which obviously cannot be the case.

So we have a finite number of points x_n on ∂D for which $A_T(x_n) = 1$. If we want to use Rouché to prove that $x - A_T(x)$ has no zeroes in $\mathring{D}$, we need a contour that at least contains the whole of $\mathring{D}$ and on which

$$|x| > |A_T(x)|. \tag{290}$$

Obviously, this inequality holds on $C = \partial D$ except for the points x_n. Hence, for each of those points we extend C with a tiny arc C_n of radius $\epsilon < \epsilon^*$ centered in x_n and outwards of D:

$$C_n = \{x : x = x_n + \epsilon e^{\jmath\alpha'}\}$$

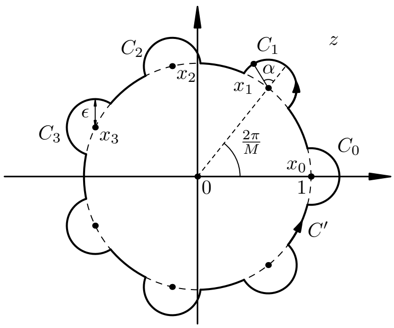


Figure 4.10: The extended contour $C' = C \setminus \{x_n, n=0,\ldots,M-1\} \cup C_0 \cup \ldots \cup C_{M-1}$ around the unit disc and all the points x_n in case $M=7$.

$$= \{x : x = e^{\jmath 2\pi n/M}(1+\epsilon e^{\jmath\alpha})\} \qquad n = 0,\ldots,M-1\,,$$

where $\alpha = \alpha' - \frac{2\pi n}{M}$ ranges from $-\frac{\pi}{2} - \frac{1}{2}\arccos(1-\frac{\epsilon^2}{2})$ to $\frac{\pi}{2} + \frac{1}{2}\arccos(1-\frac{\epsilon^2}{2})$ to join seamlessly with C (Fig. 4.10). We show that (290) holds on each contour C_n if $\epsilon \to 0$. First, on C_n, we have using Newton's binomium:

$$\begin{aligned}
A_T(e^{\jmath 2\pi n/M}(1+\epsilon e^{\jmath\alpha})) &= \sum_{h=0}^{+\infty} a_T(h)\, e^{\jmath h 2\pi n/M}\,(1+\epsilon e^{\jmath\alpha})^h \\
&= \sum_{h=0}^{+\infty} a_T(h)\, e^{\jmath h 2\pi n/M}\,\big(1 + h\,\epsilon\, e^{\jmath\alpha} + o(\epsilon)\big) \\
&= \sum_{h=0}^{+\infty} a_T(h)\, e^{\jmath h 2\pi n/M} + \epsilon \sum_{h=0}^{+\infty} a_T(h)\, h\, e^{\jmath\alpha}\, e^{\jmath h 2\pi n/M} + o(\epsilon) \\
&= 1 \,+\, \epsilon\, e^{\jmath\alpha} \sum_{h=0}^{+\infty} a_T(h)\, h\, e^{\jmath h 2\pi n/M} \,+\, o(\epsilon)\,,
\end{aligned}$$

and therefore

$$\begin{aligned}
|A_T(e^{\jmath 2\pi n/M}(1+\epsilon e^{\jmath\alpha}))| &\leqslant |1 \,+\, \epsilon\, e^{\jmath\alpha} \sum_{h=0}^{+\infty} h\, a_T(h) \,+\, o(\epsilon)| \\
&\leqslant |1 \,+\, \epsilon\, \lambda_T\, e^{\jmath\alpha} \,+\, o(\epsilon)|\,. \qquad (291)
\end{aligned}$$

On the other hand, we have on C_n:

$$|e^{\jmath 2\pi n/M}(1+\epsilon e^{\jmath\alpha})| = |1+\epsilon e^{\jmath\alpha}| \qquad (292)$$

Since $\lambda_T < 1$, we see from (291) and (292) that the requirement (290) is satisfied on C_n for small ϵ. For α in $[-\frac{\pi}{2}, \frac{\pi}{2}]$ this is obvious. For α beyond this range such as to close the contour C', to see that (291) is smaller than (292) requires a little more geometric insight, but is straightforward nonetheless.

Now we have shown that (290) holds in all points of the contour C', we can use Rouché's theorem to prove the lemma. Conform to the formulation on p. 159, let $f_1(x) = x$ and $f_2(x) = -A_T(z)$. Then clearly $|f_2(x)| < |f_1(x)|$ on the closed contour C'. Therefore, $f_1(x)$ and $f_1(x) + f_2(x)$ must have the same number of zeroes within C'. Since $f_1(x)$ obviously has only *one* such root ($x = 0$), the function $f_1(x) + f_2(x) = x - A_T(x)$ has exactly one root in the interior of C' as well. However, we *know* that this unique root can be no other than $x = 1$, which is located outside of $\mathring{D}$. This proves the lemma. □

Theorem 3. *There exists $\epsilon > 0$ such that the function $f(z)$ as given by (283) is analytic in D_ϵ under the stated conditions (284), (285), and (286).*

Proof. We simply show that $\epsilon > 0$ can be chosen such that none of the factors and terms of (283) have singularities in D_ϵ. Due to Fact 1, it is then clear that $f(z)$ is analytic in D_ϵ.

① $\dfrac{p_0}{z - A(0,z)}$

Clearly, the denominator is analytic in D_{ϵ^*} due to assumption (286). Therefore, because of Fact 2, $(z - A(0,z))^{-1}$ is also analytic in D_{ϵ^*} except for the zeroes of $z - A(0,z)$ that are poles of $(z - A(0,z))^{-1}$. However, these zeroes are *not* poles of $f(z)$ since the second factor in (283) also vanishes for $z = A(0,z)$ as can quickly be checked. Lemma 2 states that for small enough $\epsilon' < \epsilon^*$, there is only one such zero in $D_{\epsilon'}$.

② $(z-1)A(0,z)$

Both $(z-1)$ and $A(0,z)$ are analytic in D_{ϵ^*}, so no problems here.

③ $z^2 \dfrac{A(0,z) - A(A(0,z),z)}{A(0,z)}$

Again, from assumption (286) we know that $A(0,z)$ is analytic in D_{ϵ^*}, as well as $A(y,z)$ (i.e. for $y, z \in D_{\epsilon^*}$). We can invoke Fact 3 to show that $A(A(0,z),z)$ is also analytic for $z \in D_{\epsilon''}$ for some $0 < \epsilon'' < \epsilon^*$ (let $f_1(z) = A(0,z)$ and $f_2(y) = A(y,z)$), but then we have to make sure that $\{f_1(z) : z \in D_{\epsilon''}\} \subset D_{\epsilon^*}$. Lemma 1 assures that such ϵ'' exists, and we can conclude that there are no singularities of the numerator in $D_{\epsilon''}$.

Any possible zeroes of the denominator $A(0,z)$ in $\mathring{D}$ are also zeroes of the numerator so they do not introduce singularities. Note that it is impossible to have points in D_{ϵ^*} in the neighbourhood of which $A(0,z)$ is identically zero, because that would mean that $A(0,z)$ is identically zero in *all* of D_{ϵ^*}.

④ $\dfrac{1 - A(0,z)}{A(0,z) - A_T(A(0,z))}$

First, the numerator is analytic in D_{ϵ^*} and the denominator is analytic in $D_{\epsilon''}$ for some $\epsilon'' < \epsilon^*$. For the latter, this follows from Fact 3, lemma 1 ($\{A(0,z) : z \in D_{\epsilon''}\} \subset \mathring{D}$) and the fact that $A_T(z)$, being a pgf, is analytic in $\mathring{D}$.

The candidate singularities in $D_{\epsilon''}$ of this factor are the zeroes of the denominator. However, we show that there are no such zeroes in that region. Let $x = A(0,z)$, then lemma 1 states that if $z \in D_{\epsilon''}$ then $x \in \mathring{D}$. On the other hand, for $x \in \mathring{D}$ lemma 3 assures that $x - A_T(x)$ is not zero.

Clearly, for $\epsilon = \min(\epsilon', \epsilon'')$ the theorem is shown to hold. □

According to this theorem, $f(z)$ is the unique analytic continuation (Fact 5) of the pgf $P(0,z)$ from $\aleph$ to D_ϵ. In fact, one may also argue that $f(z)$ is the unique analytic continuation of $P(0,z)$ in $\aleph$ to the whole complex plane except for the singularities of $f(z)$. The exact identification of these singularities outside D_ϵ obviously depends on the nature of $A(z_1, z_2)$ but is not required for our analysis.

In the next section, we have to resolve some unknown functions emerging from the analysis of a system with multiple reservations. The approach is very similar to the one used for the model with only one reserved space in Sec. 4.1: all unknowns can be determined by appropriate substitutions of the arguments in the functional equation. Of course, this raises the same questions as have been answered here in mathematical detail: are these manipulations valid and can we always interpret the resulting expressions as fully-fledged (partial and/or multidimensional) pgfs? Although important from a theoretical viewpoint, we will not check the necessary conditions every time, since practice learns that, besides some extraordinary cases, the results are sound and applicable.

4.3 The model with multiple reservations

The First-In-First-Out (FIFO) discipline has been our reference mechanism for queues in which all packets are treated as equally important. Ideally, one wants the delay of all packets to be as low as possible, but for a given capacity of the server and a given stream of arriving packets, there will always be *some* delay. In circumstances where this delay becomes substantial, we have the problem that all packets, as they are treated the same, also have the same delay characteristics. This may lead to inefficient situations. Indeed, packets which become useless if not conveyed very fast (e.g. for real-time applications) frequently arrive at their destination too late, due to congestion largely consisting of packets for which the delay is much less important (non-real-time applications). In such cases it is sensible to step away from the usual FIFO mechanism in favour of a scheduling mechanism that recognises the packets as belonging to one of several types (or classes) and treats them differently according to the specific requirements of each type. This allows to repartition the resources of the queue in a more intelligent way.

As before, we limit our discussion to the case of two classes only: the 1-packets and the 2-packets, representing the real-time and non real-time traffic respectively. Under Absolute Priority (AP) [182, 196, 197] the 1-packets as a group experience the lowest delay possibly attainable and this at the expense of a (very) high delay for 2-packets. Whenever there are 1-packets available in the queue, one of those packets will be scheduled for service next, regardless of how many 2-packets are present and how long they have been waiting. As we have seen, granting absolute priority to the 1-packets can be a curse as much as a blessing. If the total offered load as well as the fraction of 1-packets compared to the fraction of 2-packets is relatively high, the type 2 traffic may suffer from what is known as *packet starvation*. See for instance the curves of AP for high values of λ_1/λ_T in Fig. 4.5. As there are almost always 1-packets present in the system in that case, the 2-packets get to be served very rarely, which can lead to an extremely high delay. This can all the more be perceived as unfair or inefficient operation since the delay of the *few* 2-packets is completely sacrificed for only a very small gain in the delay of the *many* 1-packets. As such, in looking to improve on FIFO, the approach of AP may be a bridge too far. For our purposes, both mechanisms serve as reference cases: whereas there is *no* delay differentiation between the two traffic types under FIFO, this difference is *maximal* under AP.

The reservation mechanism as proposed in Sec. 4.1 is a viable trade-off between FIFO and AP. As we have shown, it realises a definite differentiation in delay although it never results in packet starvation. However, as with FIFO and AP, the operation is still *fixed*, meaning that for a certain arrival process, the trade-off between the delays of both types is at some particular level predicted by the analysis we have given. If for some reason we would want to shift the trade-off even further in favour of the type 1 traffic, there is no way to do it. We have already hinted at a solution to this problem in Sec. 4.1.4. The idea is to extend the reservation mechanism to accommodate for *multiple* reserved spaces in the queue.

In the model presented here, the 1- and 2-packets behave exactly the same as in Sec. 4.1, only now we assume that there are N (>0) reserved spaces (R) in the queue instead of only one. The policy regarding the storage of the arriving packets is *exactly the same* as in the case of the queue with one reservation. Of all the packets arriving in a slot, the 1-packets are stored first, i.e. one by one they seize the most advanced R in the queue and make a new R at the end of the queue. After this, the 2-packets are stored in the usual way. The *only* difference now is that operation starts from an empty queue with N reserved spaces already present. As before, it is obvious that the number of reservations always stays the same, due to the fact that each 1-packet seizes an R but at the same time also creates an R at the end of the queue. Recall also that a reservation can *never* enter the server!

As we will see, a queue implemented with more than one reserved space is able to realise a larger differentiation of the delay. The higher we choose N, the more the delays of type 1 and type 2 packets will differ. If N becomes very large, the experienced delay is the same as that under AP. In this way, the number of reservations N in the queue can be seen as a parameter by which the *amount* of delay differentiation can be carefully controlled. This is useful from a designer's or network manager's point of view. If in some situation we feel that the 2-packets have unfairly large delays compared to the 1-packets, we can simply decrease N until a better trade-off is reached. Likewise, the delay of the 1-packets can be decreased by increasing N, at the cost of a larger delay for the 2-packets of course. In this respect, the results from the present section will prove to be very useful, since we predict the mean delay experienced by both types of packets, as well as their delay distribution for a given number of reservations in the queue.

4.3.1 Description of the model

As before, we consider a discrete-time queue operating in slotted time, where a slot is exactly the time required to serve one packet. The queue has infinite storage capacity and one server. Also, the arrival process is the same as described in Sec. 4.1.1. The number of arrivals $a_{1,k}$ of type 1 and $a_{2,k}$ of type 2 in slot k are independent from

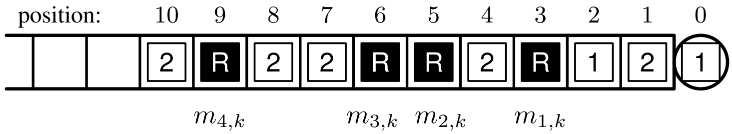


Figure 4.11: The positions of the reservations in a system with $N=4$ reserved spaces during slot k are given by the variables $m_{j,k}$ ($j=1,\ldots,4$).

slot to slot, but may be correlated *within* a slot with joint pgf $A(z_1,z_2)$, see (234). The total number of arrivals $a_{T,k}=a_{1,k}+a_{2,k}$ in slot k has pgf $A_T(z)=A(z,z)$. To keep the forthcoming expressions tractable, we extend the notation employed in Sec. 4.1.1 as follows. Let the mass function of the number of 1-arrivals per slot be

$$\beta_i \triangleq \mathrm{Prob}[a_{1,k}=i]\,, \quad i\geqslant 0\,, \qquad \text{and} \quad \alpha \triangleq \beta_0 = A_1(0) = A(0,1)\,, \tag{293}$$

such that $A_1(z)=\sum_{i=0}^{+\infty}\beta_i\, z^i$. We also define

$$A_{i*}(z) \triangleq \mathrm{E}[z^{a_{2,k}}\{a_{1,k}=i\}]\,, \qquad i\geqslant 0\,, \tag{294}$$

which is the pgf of the number of 2-arrivals in case there are exactly i arrivals of type 1 in the same slot, and where $\mathrm{E}[X\{Y\}]=\mathrm{E}[X|Y]\,\mathrm{Prob}[Y]$ as usual. Note that

$$\beta_i = A_{i*}(1)\,, \qquad i\geqslant 0\,. \tag{295}$$

As with all models we treated, we are challenged to find a suitable representation of the system state in slot k that allows us to analyse the delay distribution of the packets. If we were only interested in the equilibrium distribution of the *total* number of packets in the queue, then $\{u_k\}$ would suffice as representation. Recall that u_k is the number of packets in the queue at the beginning of slot k. Obviously, the content u_k in our multiple reservation queue is the same as for a simple GI-1-1 FIFO queue, because it is governed by (243) in both systems. The analysis of the GI-1-1 system then yields the pgf $U(z)$ of the queue content, given by (251). However, if we want to study the distribution of the *packet delay*, we require more information. As will be discussed in Sec. 4.3.4, the jth packet in an arriving batch of 1-packets seizes the jth reserved space if $j\leqslant N$. Hence, besides the queue content u_k we also need to keep track of the precise positions of all N reservations.

We number the R's from 1 to N, with the first reservation being the one *closest* to the server, i.e. the one with the smallest position number. Conversely, we may also say that the jth reservation ($j=1,\ldots,N$) has *order* j. At the start of slot k, we denote the position of the order j reservation by $m_{j,k}$, as is illustrated in Fig. 4.11. As a 1-packet always seizes the R with the lowest position number, it is seen that there can be no 1-packets in the queue on positions larger than $m_{1,k}$. In other words, all packets behind the first reservation must be of type 2.

We refer to the queue content u_k together with the positions of all R's, i.e. to the vector $\{m_{j,k}, j=1,\ldots,N; u_k\}$, as the *system state* at the start of slot k. Because of the stochastic nature of the arrival process, the system state observed in subsequent slots is an $(N+1)$-dimensional stochastic process as well. We refer to any of the random variables $m_{1,k}, m_{2,k}, \ldots, m_{N,k}; u_k$ as a *system state variable*. In order to analyse the delay distribution, we first require the equilibrium distribution of the system state itself, which will be the subject of Sec. 4.3.2. However, to obtain this distribution, we first need to construct the equations that govern the evolution of the system state from one slot to the next.

Specifically, the following system equations establish the value of the system state variables in slot $k+1$, for all possible values of those variables in slot k. The working method is to start from a certain state at the start of slot k. Then consider every possible event *during* this slot in terms of arrivals, storage, scheduling and departures and finally, write down the new system state this results in at the start of slot $k+1$. In principle, this yields a function from one $(N+1)$-dimensional space to another, although this space can be somewhat reduced by ruling out states than can never be reached. For instance, because of their physical meaning, we know that the system state variables must satisfy the constraint

$$1 \leqslant m_{1,k} < m_{2,k} < \ldots < m_{N,k} \leqslant (u_k - 1)^+ + N\,, \tag{296}$$

which for the position of jth reservation individually results in

$$j \leqslant m_{j,k} \leqslant (u_k - 1)^+ + j\,, \qquad j=1,\ldots,N\,. \tag{297}$$

Note that (242) represents the same constraints in case $j = N = 1$ since $m_k = m_{1,k}$ in the current terminology. As a convention for the remainder of this section, let j indicate any value from 1 to N, unless explicitly stated otherwise. In our analysis it turns out that, instead of the variables $m_{j,k}$, it is often more convenient to work with the variables

$$\hat{m}_{j,k} \triangleq m_{j,k} - j\,, \tag{298}$$

which all have 0 as their minimal value instead of j. Therefore, the constraints (296) and (297) now respectively become

$$0 \leqslant \hat{m}_{1,k} \leqslant \hat{m}_{2,k} \leqslant \ldots \leqslant \hat{m}_{N,k} \leqslant (u_k - 1)^+\,, \tag{299}$$

$$0 \leqslant \hat{m}_{j,k} \leqslant (u_k - 1)^+\,. \tag{300}$$

Obviously, knowledge of the value or distribution of $m_{j,k}$ implies that of $\hat{m}_{j,k}$ and vice versa, so we may interchangeably use both as system state variables.

For the system equations we can distinguish between four cases, in all of which the new queue content u_{k+1} is determined by (243). Observe also that $a_{2,k}$ appears in (243) but not in any of the following equations for $\hat{m}_{j,k+1}$ where only the number of arrivals of type 1 is of importance. Obviously, this is due to the fact that all the 1-packets arriving in slot k are stored in the queue *prior* to the 2-packets (see Fig. 4.2). Only the 1-packets seize reservations and make new ones, while the 2-packets are simply stored at the end of the queue.

Assuming that the system is not empty to begin with, the events during slot k can generally be summarised as follows. First, there are $a_{1,k}$ 1-arrivals to be stored. One by one they seize the first R and make a new one at the end. As such, seen as a group, the first N of these 1-arrivals take R's that existed before slot k, while any remaining 1-arrivals seize an R that was created by a previous 1-arrival in slot k. At the end of the slot, after the 2-packets have been stored as well, the packet in the server terminates its service and leaves the queue. Then, at the start of slot $k+1$ a new packet will enter service, at least if there are any left in the system. It is the first packet (the one with lowest position number p) that will jump over any R's at positions 1 to $p-1$ into the server at position 0. Then, since position p is free now, all packets and reservations on positions larger than p shift one position towards the server. Note that only 2-packets can actually jump over reservations into the server, while 1-packets are always positioned in *front* of any R's in the queue. Additionally, if the system is empty, we note that the first arrival (of type 1 if $a_{1,k} > 0$, of type 2 otherwise) will not enter service immediately, but has to wait until the start of the next slot. As we have said, these considerations lead us to distinguish four groups of system equations as follows.

► $u_k = 0$ (empty system)
First of all, in case of an empty system we know that the N reservations are grouped together on positions 1 to N, so

$$\begin{aligned} u_k = 0 \quad &\Rightarrow \quad m_{1,k} = 1,\, m_{2,k} = 2,\, \ldots, m_{N,k} = N \\ &\Leftrightarrow \quad \hat{m}_{1,k} = \hat{m}_{2,k} = \ldots = \hat{m}_{N,k} = 0\,. \end{aligned}$$

Therefore, we have in slot $k+1$:

$$\hat{m}_{j,k+1} = (a_{1,k} - 1)^+\,. \qquad \textbf{(Empty)}$$

► $u_k > 0, a_{1,k} = 0$ (no 1-arrivals)
In this and the remaining cases, we know that the system is not empty in slot k. Since $a_{1,k} = 0$ there are no arrivals of type 1 here. None of the reservations will be seized so they all survive to the next slot. However, after the packet in service during slot k has left, they will be shifted by one position as far as the lower constraint in (300) is not violated. We have

$$\hat{m}_{j,k+1} = (\hat{m}_{j,k} - 1)^+\,. \qquad \textbf{(Keep)}$$

► $u_k > 0, a_{1,k} = i$ with $1 \leqslant i < N$
In this case, the number of 1-arrivals i is smaller than the number of reservations N. These i arrivals seize the first i reservations, i.e. those at positions $m_{1,k}$ up to $m_{i,k}$ and make i new reservations at the end of the queue. So the *last* i reservations in the next slot will be newly created and positioned together at the end of the queue. Then, accounting for the fact that one packet will leave, it turns out that we have

$$\hat{m}_{j,k+1} = u_k + i - 2\,, \qquad \text{if} \quad j = N-i+1, \ldots, N\,. \qquad \textbf{(AtEnd)}$$

On the other hand, the *first* $N-i$ R's in the new slot are reservations that were not seized and have survived. Their ordering number has simply decreased by i or in other words, they have 'i-shifted':

$$\hat{m}_{j,k+1} = \hat{m}_{j+i,k} + i - 1\,, \qquad \text{if} \quad j = 1, \ldots, N-i\,. \qquad \textbf{(i-shift)}$$

► $u_k > 0, a_{1,k} = i$ with $i \geqslant N$
Now, there are at least as many 1-arrivals as there are reservations. In this case all new reservations will be grouped at the end of the queue, since none of the old R's survive. The equation **(AtEnd)** now applies for *all* new reservation positions, i.e.

$$\hat{m}_{j,k+1} = u_k + i - 2\,, \qquad j = 1, \ldots, N\,. \tag{AtEnd}$$

4.3.2 Equilibrium distribution of the system state

Now we use the above equations to obtain the distribution of the system state $\{m_{j,k}, j = 1, \ldots, N; u_k\}$ for $k \to \infty$, assuming that the system reaches equilibrium. To this end, let us define the joint pgf of the system state at the start of slot k as

$$\begin{aligned} P_k(y_1, y_2, \ldots, y_N; z) &\triangleq \mathrm{E}[y_1^{\hat{m}_{1,k}} y_2^{\hat{m}_{2,k}} \cdot \cdots \cdot y_N^{\hat{m}_{N,k}} z^{u_k}] \\ &= \mathrm{E}[y_1^{m_{1,k}-1} y_2^{m_{2,k}-2} \cdot \cdots \cdot y_N^{m_{N,k}-N} z^{u_k}]\,. \end{aligned} \tag{301}$$

The equations **(Empty)**, **(Keep)**, **(*i*-shift)** and **(AtEnd)** allow to relate the joint pgf P_{k+1} of the system state in slot $k+1$ to the distribution P_k in slot k. We perform separate calculations for the four cases above, splitting up the joint pgf into four terms as

$$\begin{aligned} P_{k+1}(y_1, y_2, \ldots, y_N; z) &= \mathrm{E}[y_1^{\hat{m}_{1,k+1}} y_2^{\hat{m}_{2,k+1}} \cdot \cdots \cdot y_N^{\hat{m}_{N,k+1}} z^{u_{k+1}}] \\ &= \mathrm{E}[\ldots\{u_k = 0\}] + \mathrm{E}[\ldots\{u_k > 0, a_{1,k} = 0\}] \\ &\quad + \sum_{i=1}^{N-1} \mathrm{E}[\ldots\{u_k > 0, a_{1,k} = i\}] + \sum_{i=N}^{+\infty} \mathrm{E}[\ldots\{u_k > 0, a_{1,k} = i\}]\,. \end{aligned} \tag{302}$$

For the first term, **(Empty)** applies, as well as (243) such that

$$\begin{aligned} &\mathrm{E}[y_1^{\hat{m}_{1,k+1}} y_2^{\hat{m}_{2,k+1}} \cdot \cdots \cdot y_N^{\hat{m}_{N,k+1}} z^{u_{k+1}}\{u_k = 0\}] \\ &\quad = \mathrm{E}[(y_1\, y_2\, \cdots\, y_N)^{(a_{1,k}-1)^+}\, z^{a_{1,k}+a_{2,k}}\{u_k = 0\}] \\ &\quad = \ldots \\ &\quad = \frac{p_{0,k}}{y_1\, y_2\, \cdots\, y_N}\big[(y_1\, y_2\, \cdots\, y_N - 1)A(0,z) + A(y_1\, y_2\, \cdots\, y_N\, z, z)\big]\,, \end{aligned} \tag{303}$$

where, analogous to (247),

$$p_{0,k} = U_k(0) = P_k(0, 0, \ldots, 0; 0)\,, \tag{304}$$

is the probability that the system is empty at the beginning of slot k. The second term of (302) can be further developed with **(Keep)**, which yields

$$\begin{aligned} &\mathrm{E}[y_1^{\hat{m}_{1,k+1}} y_2^{\hat{m}_{2,k+1}} \cdots y_N^{\hat{m}_{N,k+1}} z^{u_{k+1}}\{u_k > 0, a_{1,k} = 0\}] \\ &\quad = \mathrm{E}[y_1^{(\hat{m}_{1,k}-1)^+} y_2^{(\hat{m}_{2,k}-1)^+} \cdots y_N^{(\hat{m}_{N,k}-1)^+} z^{u_k-1+a_{2,k}}\{u_k > 0, a_{1,k} = 0\}] \\ &\quad = A(0,z)\mathrm{E}[y_1^{(\hat{m}_{1,k}-1)^+} y_2^{(\hat{m}_{2,k}-1)^+} \cdots y_N^{(\hat{m}_{N,k}-1)^+} z^{u_k-1}\{u_k > 0\}] \end{aligned}$$

$$
\begin{aligned}
&= A(0,z)\sum_{n=1}^{+\infty}\sum_{j_1=0}^{n-1}\sum_{j_2=j_1}^{n-1}\sum_{j_3=j_2}^{n-1}\cdots\sum_{j_N=j_{N-1}}^{n-1} y_1^{(j_1-1)^+}\cdots y_N^{(j_N-1)^+} z^{n-1} p_k(j_1,\dots,j_N;n)\\
&= A(0,z)\sum_{n=1}^{+\infty} z^{n-1}\Bigg[\\
&\qquad \sum_{j_1=1}^{n-1}\sum_{j_2=j_1}^{n-1}\sum_{j_3=j_2}^{n-1}\sum_{j_4=j_3}^{n-1}\cdots\sum_{j_N=j_{N-1}}^{n-1} y_1^{j_1-1}y_2^{j_2-1}y_3^{j_3-1}\cdots y_N^{j_N-1}p_k(j_1,\dots,j_N;n)\\
&\qquad\quad +\sum_{j_2=1}^{n-1}\sum_{j_3=j_2}^{n-1}\sum_{j_4=j_3}^{n-1}\cdots\sum_{j_N=j_{N-1}}^{n-1} y_2^{j_2-1}y_3^{j_3-1}\cdots y_N^{j_N-1}p_k(0,j_2,\dots,j_N;n)\\
&\qquad\qquad +\sum_{j_3=1}^{n-1}\sum_{j_4=j_3}^{n-1}\cdots\sum_{j_N=j_{N-1}}^{n-1} y_3^{j_3-1}\cdots y_N^{j_N-1}p_k(0,0,j_3,\dots,j_N;n)\\
&\qquad\qquad\qquad +\dots\\
&\qquad\qquad\qquad\quad +\sum_{j_N=1}^{n-1} y_N^{j_N-1}p_k(0,0,\dots,0,j_N;n)\\
&\qquad\qquad\qquad\qquad +p_k(0,0,\dots,0;n)\Bigg]\\
&= \frac{A(0,z)}{z}\Bigg[\frac{1}{y_1y_2y_3\cdots y_N}\Big(P_k(y_1,y_2,y_3,\dots,y_N;z)-P_k(0,y_2,y_3,\dots,y_N;z)\Big)\\
&\qquad +\frac{1}{y_2y_3\cdots y_N}\Big(P_k(0,y_2,y_3,\dots,y_N;z)-P_k(0,0,y_3,\dots,y_N;z)\Big)\\
&\qquad\quad +\dots\\
&\qquad\qquad +\frac{1}{y_N}\Big(P_k(0,0,\dots,0,y_N;z)-P_k(0,0,\dots,0,0;z)\Big)\\
&\qquad\qquad\quad +P_k(0,0,\dots,0,0;z)-p_{0,k}\Bigg],
\end{aligned}
\tag{305}
$$

where we have used the following notation for the mass function of the system state distribution in slot k:

$$
p_k(j_1,j_2,\dots,j_N;n)\triangleq \mathrm{Prob}[\hat{m}_{1,k}=j_1,\ \hat{m}_{2,k}=j_2,\ \dots,\hat{m}_{N,k}=j_N,\ u_k=n]. \tag{306}
$$

In the third term of (302), we must apply **(i-shift)** for the new reservations of order 1 to $N-i$ and **(AtEnd)** for the remaining reservation positions. We have for $1\leqslant i<N$, using (294):

$$
\begin{aligned}
&\mathrm{E}[\underbrace{y_1^{\hat{m}_{1,k+1}}\cdots y_{N-i}^{\hat{m}_{N-i,k+1}}}_{\textbf{(i-shift)}}\cdot\underbrace{y_{N-i+1}^{\hat{m}_{N-i+1,k+1}}\cdots y_N^{\hat{m}_{N,k+1}}}_{\textbf{(AtEnd)}}\cdot z^{u_{k+1}}\{u_k>0,a_{1,k}=i\}]\\
&=\mathrm{E}[y_1^{\hat{m}_{i+1,k}+i-1}\cdots y_{N-i}^{\hat{m}_{N,k}+i-1}\,y_{N-i+1}^{u_k+i-2}\cdots y_N^{u_k+i-2}z^{u_k-1+i+a_{2,k}}\{u_k>0,a_{1,k}=i\}]\\
&=(zy_1y_2\cdots y_{N-i})^{i-1}(y_{N-i+1}\cdots y_N)^{i-2}A_{i*}(z)
\end{aligned}
$$

$$\begin{aligned}
&\mathrm{E}[y_1^{\hat{m}_{i+1,k}} \cdots y_{N-i}^{\hat{m}_{N,k}} (zy_{N-i+1} \cdots y_N)^{u_k} \{u_k > 0\}] \\
&= (zy_1y_2 \cdots y_{N-i})^{i-1} (y_{N-i+1} \cdots y_N)^{i-2} A_{i*}(z) \\
&\quad \big[P_k(\underbrace{1, 1, \ldots, 1}_{i}, y_1, y_2, \ldots, y_{N-i}; zy_{N-i+1} \cdots y_N) - p_{0,k}\big] . \qquad (307)
\end{aligned}$$

Finally, in the last term of (302), we find for $i \geqslant N$ and using **(AtEnd)**,

$$\begin{aligned}
&\mathrm{E}[y_1^{\hat{m}_{1,k+1}} y_2^{\hat{m}_{2,k+1}} \cdots y_N^{\hat{m}_{N,k+1}} z^{u_{k+1}} \{u_k > 0, a_{1,k} = i\}] \\
&\qquad = z^{i-1} (y_1y_2 \cdots y_N)^{i-2} A_{i*}(z) \big[U_k(zy_1y_2 \cdots y_N) - p_{0,k}\big] , \qquad (308)
\end{aligned}$$

where $U_k(z)$ is the marginal pgf of the queue content u_k. Adding up the terms (303), (305), (307) and (308) yields the right-hand side of (302). If equilibrium kicks in for $k \to \infty$, the distributions P_{k+1} and P_k become identical. In other words, all distributions and probabilities become independent of the slot index k. Therefore, this index may be dropped to indicate that we refer to an arbitrary slot during equilibrium. If we single out the function $P(y_1, \ldots, y_N; z)$, we find our basic functional equation for the equilibrium distribution of the system state in a queue with N reservations:

$$\begin{aligned}
&\big(z\,\tilde{y}_1 - A(0, z)\big) P(y_1, y_2, \ldots, y_N; z) \\
&\quad = z\, p_0 \Big[(\tilde{y}_1 - 1) A(0, z) + A(z\,\tilde{y}_1, z)\Big] \\
&\qquad + A(0, z) \Big[\, (y_1 - 1)\, P(0, y_2, \ldots, y_N; z) \\
&\qquad\qquad + y_1(y_2 - 1)\, P(0, 0, y_3, \ldots, y_N; z) \\
&\qquad\qquad + y_1y_2(y_3 - 1)\, P(0, 0, 0, y_4, \ldots, y_N; z) \\
&\qquad\qquad + \ldots \\
&\qquad\qquad + y_1y_2 \cdots y_{N-1}(y_N - 1)\, P(0, 0, \ldots, 0; z) - \tilde{y}_1 p_0 \Big] \\
&\qquad + \sum_{i=1}^{N-1} \frac{(z\,\tilde{y}_1)^i}{\tilde{y}_{N-i+1}} A_{i*}(z) \Big[P(\underbrace{1, \ldots, 1}_{i}, y_1, y_2, \ldots, y_{N-i}; z\,\tilde{y}_{N-i+1}) - p_0\Big] \\
&\qquad + \sum_{i=N}^{+\infty} z^i \tilde{y}_1^{i-1} A_{i*}(z) \Big[U(z\,\tilde{y}_1) - p_0\Big] , \qquad (309)
\end{aligned}$$

where we define $\tilde{y}_j$ as the following product:

$$\tilde{y}_j \triangleq y_j \cdot y_{j+1} \cdot \cdots \cdot y_N . \qquad (310)$$

This functional equation completely determines the equilibrium distribution P, although we see that a lot of unknown functions have yet to be determined. Nevertheless, all these unknowns can be resolved by using relation (309) only, as we will demonstrate. Observe that there are a total of $2N$ unknown functions on the right-hand side of (309). Let us designate a shorthand to each of these functions and order them in a list as follows

1. $\boxed{1} = P(0, y_2, y_3, y_4, \ldots, y_N; z)$

$$
\begin{array}{rl}
2. & \colorbox{black}{\color{white}1} = P(1, y_2, y_3, y_4, \ldots, y_N; z)\\
3. & \boxed{2} = P(0, 0, y_3, y_4, \ldots, y_N; z)\\
4. & \colorbox{black}{\color{white}2} = P(1, 1, y_3, y_4, \ldots, y_N; z)\\
5. & \boxed{3} = P(0, 0, 0, y_4, \ldots, y_N; z)\\
6. & \colorbox{black}{\color{white}3} = P(1, 1, 1, y_4, \ldots, y_N; z)\\
 & \vdots\\
2N-1. & \boxed{N} = P(0, 0, 0, 0, \ldots, 0; z)\\
2N. & \colorbox{black}{\color{white}N} = P(1, 1, 1, 1, \ldots, 1; z) = U(z)
\end{array}
\tag{311}
$$

Note that we could have included the probability p_0 in this list as well, although it easily follows as $p_0 = 1 - \lambda_T$ by imposing the normalisation condition on $U(z)$. The unknown functions can be determined in this order by performing the appropriate substitutions in (309). In fact, the functional equation is able to provide each of the unknowns in the list as a function of unknowns *further* in the list. To make clear how this is done, let us denote by $\blacksquare$ the function $P(y_1, y_2, \ldots, y_N; z)$ in an *explicit* form, i.e. equal to (309) but with all unknowns (those in list (311)) resolved. On the other hand, we represent by $\blacksquare \;?\; \boxed{1}\colorbox{black}{\color{white}1}\boxed{2}\colorbox{black}{\color{white}2}\ldots\boxed{N}\colorbox{black}{\color{white}N}$ a relation determining $P(y_1, y_2, \ldots, y_N; z)$, but in which the functions after the question mark are still unresolved. Obviously, this is the functional equation in the form given by (309). Clearly, the final explicit expression for the equilibrium distribution of the system state we are looking for is $\blacksquare$. Even for small N though, obtaining $\blacksquare$ is an enormous task to do by hand, so we only explain *how* to do this, rather than actually doing it.

As we have said, (309) holds the key to determining all the unknown functions explicitly by evaluating it for the right arguments. In what follows, we describe a 'binary tree backtracking' scheme that shows us the way. There are *two types* of substitutions that yield relevant information. The first one is to let

$$
y_1 \to 1,\; y_2 \to 1,\; \ldots,\; y_{n-1} \to 1,\; y_n \to 1\,, \tag{312}
$$

for some $n = 1, \ldots, N$. This directly gives the relation $\colorbox{black}{\color{white}n} \;?\; \boxed{n{+}1}\colorbox{black}{\color{white}$n{+}1$}\ldots\boxed{N}\colorbox{black}{\color{white}N}$. The second type of substitution is to let

$$
y_1 \to 1,\; y_2 \to 1,\; \ldots,\; y_{n-1} \to 1,\; y_n \to \frac{A(0,z)}{y_{n+1} y_{n+2} \cdots y_N z}\,, \tag{313}
$$

which is mostly the same as (312), except for the last step. Note that if (313) is performed on (309), the left-hand side vanishes. Using a similar argumentation as on p. 149 where we determined $\boxed{1} = P(0, z)$, we know that the right-hand side has to vanish then as well. This provides the relation $\boxed{n} \;?\; \colorbox{black}{\color{white}n}\boxed{n{+}1}\colorbox{black}{\color{white}$n{+}1$}\ldots\boxed{N}\colorbox{black}{\color{white}N}$.

We can arrange the substitutions of type (312) and (313) in a *binary tree* as shown on the left side of Fig. 4.12. Branches going down correspond to substitutions $y_j \to 1$, while branches to the right indicate a substitution $y_j \to A(0, z)/\tilde{y}_{j+1} z$. Hence, every path in this tree corresponds to either (312) or (313), depending on the last branch. In other words, each path represents a sequence of substitutions which, if applied to the functional equation, yield the relation indicated on the node the path ends in.

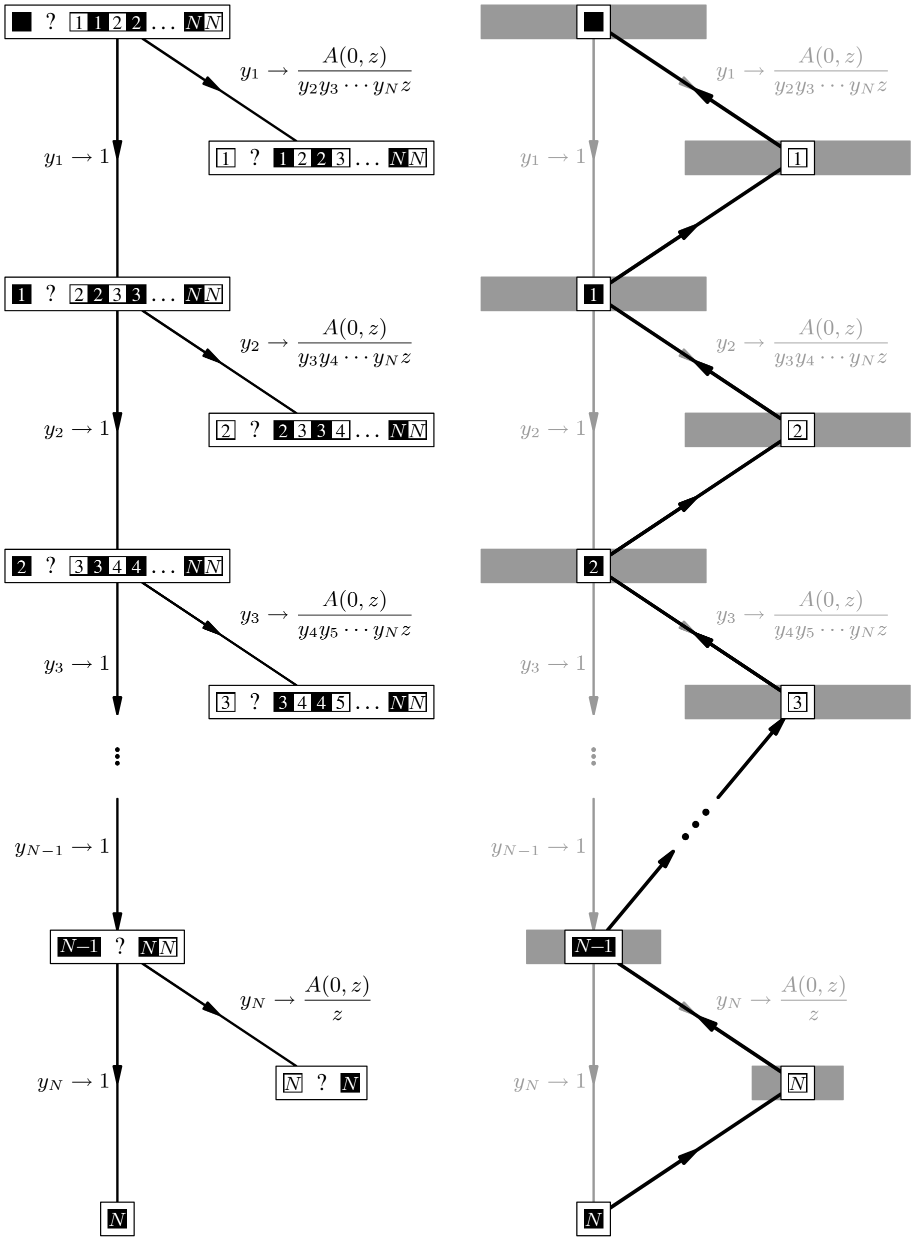


Figure 4.12: Binary tree backtracking scheme to obtain the unknown functions.

Starting from the top of the tree, we can progressively determine the relations on each node, until finally, we obtain $\colorbox{black}{\textcolor{white}{N}} = U(z)$ explicitly. Note that we have included the latter function in the list of unknowns notwithstanding the fact that it can easily be obtained from the GI-1-1 analysis. We did so to illustrate that the marginal process of the queue content is 'contained' within that of the system with multiple reservations. So, although we *can* rely on the results of GI-1-1, we do not have to in theory.

On the right side of Fig. 4.12, we show the second part of the calculation scheme. Starting from the bottom node, we work our way to the top by backtracking the previously obtained unresolved relations. Indeed, once we have $\colorbox{black}{\textcolor{white}{N}}$, we can use this in the node with relation $\boxed{N}$? $\colorbox{black}{\textcolor{white}{N}}$ to resolve $\boxed{N}$. In turn, the explicit expressions $\colorbox{black}{\textcolor{white}{N}}$ and $\boxed{N}$ allow to obtain $\colorbox{black}{\textcolor{white}{$N{-}1$}}$ in the node with relation $\colorbox{black}{\textcolor{white}{$N{-}1$}}$? $\boxed{N}\colorbox{black}{\textcolor{white}{N}}$, and so forth until we reach the top. At this point, we have an explicit expression for $\blacksquare$, as well as for every other unknown in the list (311). Note that in case $N = 1$, the scheme exists of only one stage containing substitutions $y_1 \to 1$ and $y_1 \to A(0,z)/z$, which we have used in Sec. 4.1 to obtain $U(z)$ and $P(0,z)$ respectively.

4.3.3 A basic theorem

We now discuss an important property regarding the behaviour of systems with multiple reservations which may not readily be apparent from the analysis so far. Let us first introduce the following notation: $m_{j,k}^{[N]}$ is the position of the jth reservation at the beginning of slot k in a system with N reservations, or a NR-system as we will call it. Corresponding to (298), let also $\hat{m}_{j,k}^{[N]} = m_{j,k}^{[N]} - j$. Of course, if it is clear from context that we are considering a system with N reservations, the superscript $[N]$ may be dropped. In what follows, we use this notation for other quantities as well, to indicate the number of reservations in the system they are related to. The following theorem is crucial to our analysis of the packet delay distribution.

Theorem 4 (Reservation Theorem, RT)**.** *If a queue with N reservations and a queue with $N-1$ reservations are both empty in slot 0 and are both subjected to the same number of arriving 1- and 2-packets in each of the following slots 0 to k, then we have that*

$$m_{j,k}^{[N]} = m_{j-1,k}^{[N-1]} + 1\,, \quad \textit{or equivalently,} \quad \hat{m}_{j,k}^{[N]} = \hat{m}_{j-1,k}^{[N-1]}\,, \tag{314}$$

at the beginning of slot k, for $j = 2, \ldots, N$.

Proof. As we assume that both systems are empty in slot 0, the variables $\hat{m}_j^{[N]}$ and $\hat{m}_j^{[N-1]}$ are all equal to 0 such that (314) holds. Now, suppose that (314) holds in slot k for all $j = 2, \ldots, N$. If we can show that

$$\hat{m}_{j,k+1}^{[N]} = \hat{m}_{j-1,k+1}^{[N-1]}\,, \qquad j = 2, \ldots, N\,, \tag{315}$$

then by induction, this proves the theorem. For certain values of the reservation positions in slot k, the system equations **(Empty)**, **(Keep)**, **(AtEnd)** and **(i-shift)** provide the new reservation positions in slot $k+1$. Therefore, we must compare these equations to their equivalent in case of a system with only $N-1$ reservations. Doing so, assuming that (314) holds, it can be checked easily that (315) holds as well, for every possible value of u_k and $a_{1,k}$. $\square$

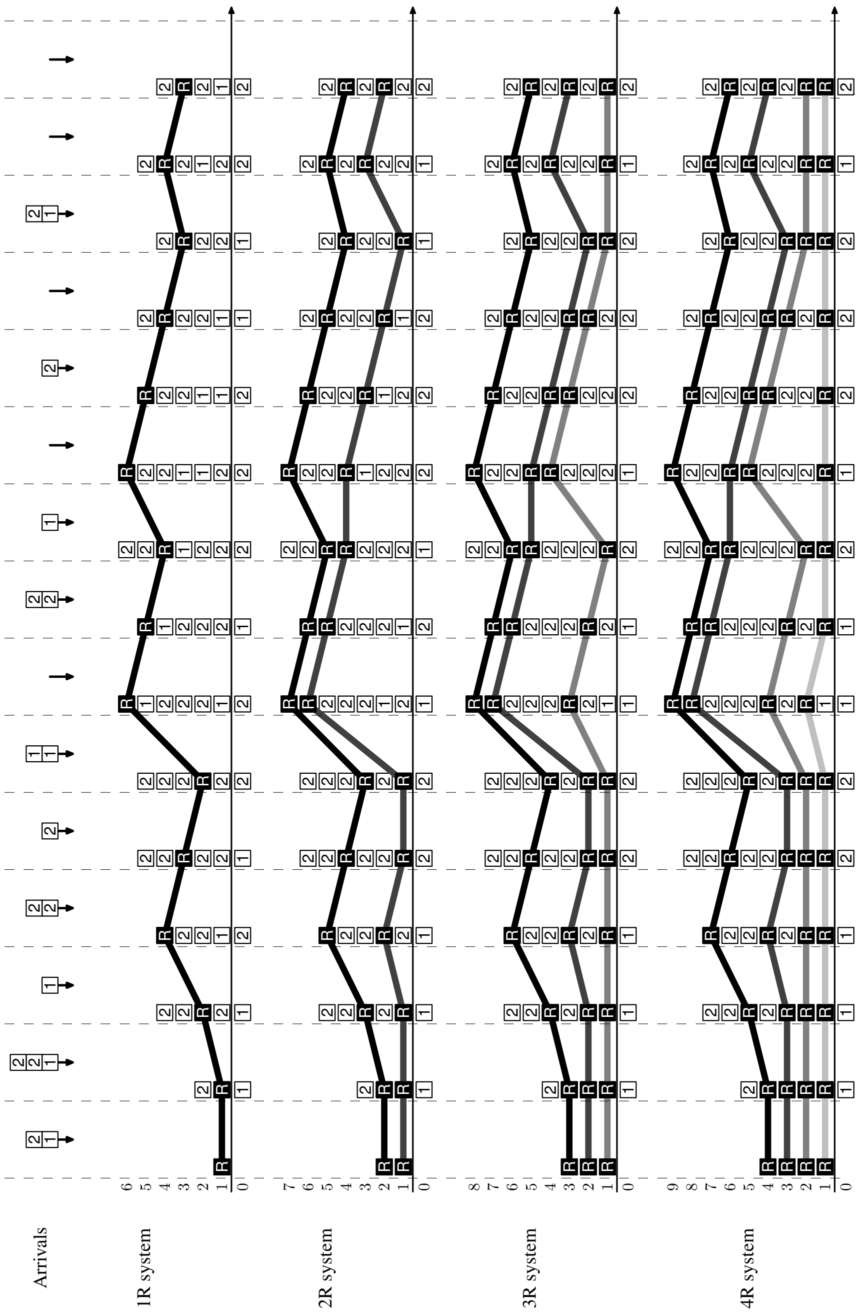


Figure 4.13: Illustration of the Reservation theorem

This theorem is illustrated in Fig. 4.13, where we show the evolution of the queue content starting from an empty system in the first slot and for the specific trace of arrivals given on the top. For four different systems with respectively 1, 2, 3 and 4 reserved spaces, the queue content is indicated at the start of each slot. To make our point clear, we connected the reservations of the same order with a bold line that is darker for reservations of higher orders. The relation (314) is clearly seen to hold. For instance, the shape of the black line connecting the positions of the first (and only) R in the 1R-system is the same as the shape of the line connecting the second R in the 2R-system, the third R in the 3R-system and so forth. The only difference is that these lines are shifted by one position. Likewise, the dark gray line indicating the position of the first reservation in the 2R-system has the same shape as that of the second reservation in the 3R-system, the third reservation in the 4R-system and so forth, again shifted by one position each time.

Together with the fact that the systems with N and $N-1$ reservations also hold the same total number of packets, theorem (314) leads to the following corollary concerning the joint pgfs of the system state of both systems. Let $P_k^{[N]}$ be the system state distribution (301) for a system with N reserved spaces. It is now easily seen that

$$\begin{aligned} P_k^{[N]}(1, y_2, \dots, y_N; z) &= \mathrm{E}[y_2^{\hat{m}_{2,k}^{[N]}} y_3^{\hat{m}_{3,k}^{[N]}} \cdot \dots \cdot y_N^{\hat{m}_{N,k}^{[N]}} z] \\ &= \mathrm{E}[y_2^{\hat{m}_{1,k}^{[N-1]}} y_3^{\hat{m}_{2,k}^{[N-1]}} \cdot \dots \cdot y_N^{\hat{m}_{N-1,k}^{[N-1]}} z] \\ &= P_k^{[N-1]}(y_2, \dots, y_N; z) \,. \end{aligned} \tag{316}$$

If we let the arguments y_2 to y_n assume the value 1, then it directly follows from (316) for some $0 \leqslant n < N$ that

$$P_k^{[N]}(1, \dots, 1, y_{n+1}, \dots, y_N; z) = P_k^{[N-n]}(y_{n+1}, \dots, y_N; z) \,. \tag{317}$$

This property says that the distribution of the last $N-n$ reservation positions in a system with N reservations is equal (up to a fixed shift) to that of reservation positions in a system with only $N-n$ reservations. Property (317) is a fundamental observation for queues with multiple reservations and we will use it abundantly in what follows.

4.3.4 Delay of type 1 packets

So far, we concentrated on the equilibrium distribution of the system state during an arbitrary slot in a system with $N \geqslant 1$ reserved spaces. This resulted in a functional equation for the joint distribution of the positions of each reservation and the total number of packets in the queue. However, as we have explained, the expression for this distribution is very hard to obtain explicitly, even for small N, as it is the result of many subsequent substitutions that interact in an untractable way. For instance, the expression for $P(y_1, y_2, y_3; z)$ could not be printed here on one page. Fortunately, from a practical point of view, we are not so much interested in the probability of such exotic events as the fifth reservation being at position 9 while the second is at position 5. The distribution of such events *could* in principle be obtained from the complete $(N+1)$-dimensional system state distribution, but then we are faced with the same problem of increasing complexity. Instead, we are more interested in a direct, useful performance measure such as the distribution of the packet delay. The analysis of the delay distribution relies heavily on our results regarding the system state distribution. However, as will be shown, we are able to circumvent some of the complexity issues and obtain numerical algorithms for the mean and the tail distribution of the packet delay, that are relatively easy to implement.

Here, we focus on the delay distribution of the type 1 packets only, deferring the analysis for the type 2 packets to Sec. 4.3.5. As in Sec. 4.1.3, let us consider an arbitrary packet $\mathcal{P}$ of type 1 again and refer to the slot in which it arrives as slot I. Our purpose now is to obtain the delay d_1 experienced by $\mathcal{P}$ as it goes through the system with N reserved spaces. Recall that the delay d_1 is the integer number of slots from the moment that $\mathcal{P}$ is stored in the queue at the end of slot I, until the moment $\mathcal{P}$ leaves the queue at the end of its service. In other words, d_1 is the time the packet $\mathcal{P}$ has to *wait* in the queue and is *being served*. Obviously, the delay of $\mathcal{P}$ will depend on the state of the system in slot I. Assuming equilibrium behaviour, we have explained in Sec. 4.1 that the system state in slot I has the same distribution as in *any* slot. Therefore, we can use (309) to provide the distribution of the reservation positions in slot I. With regard to the number of 1- and 2-arrivals during slot I, expression (257) still holds as well as (258) for the pgf of ℓ_1, which is the counting number of $\mathcal{P}$ in the batch of a_1^I arriving 1-packets. Let us also introduce

$$\omega_j \triangleq \text{Prob}[\ell_1 = j] = \frac{1}{\lambda_1}\sum_{i=j}^{+\infty}\beta_i\,, \qquad j \geqslant 1\,, \tag{318}$$

for the mass function of ℓ_1, such that according to (258),

$$L_1(z) = \sum_{j=1}^{+\infty}\omega_j\, z^j = \frac{z(1-A_1(z))}{\lambda_1(1-z)}\,. \tag{319}$$

Recall that β_i defined by (293) is the probability of having i 1-arrivals in a slot. In the following, we drop the time index in the notation of the system state variables, since it is clear they all refer to slot I.

The pgf $D_1(z)$ of the type 1 packet delay

As we have said, the delay of $\mathcal{P}$ depends on the state of the system in slot I. More specifically, what is of importance is the exact position in which $\mathcal{P}$ will be stored at the end of slot I. Only then it is clear how many packets are stored *in front* of $\mathcal{P}$, which obviously are also the packets that will be served prior to $\mathcal{P}$. In other words, the delay is the number of slots required to serve each packet positioned in front of $\mathcal{P}$ at the end of slot I, and $\mathcal{P}$ itself. If there are no packets in front of $\mathcal{P}$ when it is stored, then $\mathcal{P}$ will be served immediately in the next slot and $d_1 = 1$. So the information we must derive from the system state in slot I is the position in the queue that $\mathcal{P}$ will occupy. Note that we can ignore the 2-arrivals in slot I since they are stored later than the 1-arrivals, according to the reservation protocol. Now, if $\mathcal{P}$ is the first of the batch ($\ell_1 = 1$) it will seize the first reservation at position m_1, if $\ell_1 = 2$ then it takes position

m_2, and so forth. If ℓ_1 is larger than N however, $\mathcal{P}$ will seize a reservation created by one of the $\ell_1 - 1$ previous 1-arrivals in slot I, located somewhere at the end of the queue. All in all, we find that

$$d_1 = \begin{cases} m_j & \text{if } \ell_1 = j \leqslant N\,, \\ (u-1)^+ + \ell_1 & \text{if } \ell_1 > N\,, \end{cases} \tag{320}$$

which obviously is the extension of (259) to a system with N reservations instead of only one. Taking the z-transform and using (318), this yields for the pgf $D_1(z)$ of the delay d_1:

$$\begin{aligned} D_1(z) &= \sum_{j=1}^{N} \omega_j \mathrm{E}[z^{m_j}] \;+\; \mathrm{E}[z^{(u-1)^+}] \sum_{j=N+1}^{+\infty} \omega_j z^j \\ &= \sum_{j=1}^{N} \omega_j \mathrm{E}[z^{\hat{m}_j}] z^j \;+\; \mathrm{E}[z^{(u-1)^+}] \Big(L_1(z) - \sum_{j=1}^{N} \omega_j z^j \Big)\,, \end{aligned} \tag{321}$$

where we have used the alternative representation (298) for the jth reservation position in slot I. From (251), we easily find that

$$\mathrm{E}[z^{(u-1)^+}] = p_0 \frac{z-1}{z - A_T(z)}\,. \tag{322}$$

The marginal distribution $\mathrm{E}[z^{\hat{m}_j}]$ appearing in (321) on the other hand, is more difficult to obtain. However, this is where theorem 4 and its corollary (317) come into play. Let us explicitly indicate the number of reservations in the system to which a certain variable corresponds. Then we have for $j = 1, \ldots, N$:

$$\begin{aligned} \mathrm{E}[z^{\hat{m}_j^{[N]}}] &= P^{[N]}(\underbrace{1, 1, \ldots, 1}_{j-1}, z, 1, 1, \ldots, 1; 1) \\ &= P^{[N-j+1]}(z, 1, 1, \ldots, 1; 1) \;=\; \mathrm{E}[z^{\hat{m}_1^{[N-j+1]}}]\,. \end{aligned} \tag{323}$$

The conclusion is that instead of having to calculate the marginal distributions of all reservation positions $\hat{m}_j^{[N]}$ ($j = 1, \ldots, N$) in the NR-system, it is sufficient to obtain the marginal distributions of $\hat{m}_1^{[N-j+1]}$, i.e. only of *the first reservation positions* in the systems with 1 up to N reservations. The advantage obviously is that our results are better scalable as N increases: once we have the distribution of the first reservation position in a system with n reservations, that result can be reused immediately in the analysis of the delay in a system with $n+1$ reservations, which in turn can be used in case of n+2 reservations and so forth. Therefore, we shall first focus on the distribution of $\hat{m}_1$ in case of N reservations in the system.

Distribution of the first reservation position m_1

The marginal distribution of $\hat{m}_1$ can be obtained from the *full* system state distribution in slot I determined by functional equation (309). We derive a recursive relation for

$\mathrm{E}[z^{\hat{m}_1}]$ by using substitutions similar to those in the first stage of Fig. 4.12. Specifically, let all arguments in (309) be equal to 1 except for the first one, for which we take $y_1 \to z$. Using (293)–(295) and theorem 4 again in the form of (323), we find

$$(z-\alpha)P^{[N]}(z,1,1,\dots,1;1) \;=\; p_0(z-1)f_N(z) \;+\; \alpha(z-1)P^{[N]}(0,1,\dots,1;1) + \sum_{i=1}^{N-1} \beta_i z^i P^{[N-i]}(z,1,1,\dots,1;1)\,, \tag{324}$$

where

$$f_n(z) \triangleq \frac{1}{z-A_T(z)} \sum_{i=n}^{+\infty} \beta_i z^i\,, \qquad n \geqslant 1\,. \tag{325}$$

Note that the factor $(z-A_T(z))^{-1}$ is entirely due to the last term of (309) where $U(z)$ appears under the mentioned substitution. Let us also define

$$\Phi_n(z) \triangleq f_n(z) - f_n(\alpha) = \frac{1}{z-A_T(z)} \sum_{i=n}^{+\infty} \beta_i z^i - \frac{1}{\alpha-A_T(\alpha)} \sum_{i=n}^{+\infty} \beta_i \alpha^i\,. \tag{326}$$

In (324), the probability $P(0,1,\dots,1;1)$ that $\hat{m}_1 = 0$ can be obtained from the functional equation by evaluating it for the right arguments. First, let $z \to \alpha$ in (324) such that the left-hand side vanishes. As we have explained before, the other side must be equal to 0 then as well, which results in

$$P^{[N]}(0,1,\dots;1) = -\frac{p_0}{\alpha} f_N(\alpha) + \frac{1}{\alpha(1-\alpha)} \sum_{i=1}^{N-1} \beta_i \alpha^i P^{[N-i]}(\alpha,1,\dots;1)\,. \tag{327}$$

Plugging this into (324) yields the desired recursive relation for the distribution of the first reservation position. If we first introduce a shorter notation for this distribution:

$$\Omega_n(z) \triangleq P^{[n]}(z,1,\dots,1;1) = \mathrm{E}[z^{\hat{m}_1^{[n]}}]\,, \qquad n \geqslant 1\,, \tag{328}$$

then we finally find

$$\Omega_N(z) = p_0(z-1)\frac{\Phi_N(z)}{z-\alpha} + \sum_{i=1}^{N-1} \beta_i \frac{z^i \Omega_{N-i}(z) - \frac{z-1}{\alpha-1}\alpha^i \Omega_{N-i}(\alpha)}{z-\alpha}\,. \tag{329}$$

In principle, our work is done now, since this relation determines all $\Omega_n(z)$, $n = 1,\dots,N$, being the distributions of the first reservation position in systems with 1 up to N reservations. Through (321) and (323), this directly provides the pgf of the type 1 packet delay in these systems.

Indeed, (329) can be solved in an iterative way since $\Omega_N(z)$ appears only on the left-hand side while the other side only depends on $\Omega_1(z)$ to $\Omega_{N-1}(z)$. We start with $N=1$ in the first iteration which yields $\Omega_1(z)$, the next iteration for $N=2$ gives $\Omega_2(z)$, and so forth. The problem is however that we also have to determine the constant $\Omega_n(\alpha)$ in step n ($n=1,\dots,N-1$). Since $\Omega_n(z)$ is a pgf and $\alpha < 1$, we know there must be a solution for the quantities $\Omega_n(\alpha)$ lying between 0 and 1, but obtaining these values

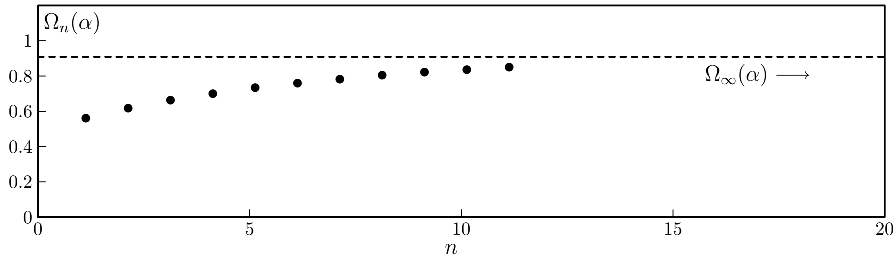


Figure 4.14: Values of $\Omega_n(\alpha)$ obtained by iteration of (329) and calculating the limit for $z \to \alpha$.

in a direct analytic way proves to be difficult. The main issue here is that if (329) is solved by iteration, the complexity doubles with each step, producing expressions of exponentially increasing length. Moreover, once we have $\Omega_n(z)$, the quantity $\Omega_n(\alpha)$ must be determined by taking the limit $z \to \alpha$ of (329) (with $N = n$ of course). It turns out that we require the n-fold use of de l'Hôpital's rule, meaning that one has to differentiate an already extensive expression n times and evaluate it for $z = \alpha$.

This iterative procedure was implemented in Maple, a symbolic computer algebra system, and we found that even for a simple arrival process as in (282), the CPU and memory requirements become unreasonably large for N as small as 10 or 12. For $\lambda_1 = \lambda_2 = 0.45$, we plotted the result in Fig. 4.14. The limiting value $\Omega_\infty(\alpha)$ was obtained by assuming decoupling in the queue, as will be explained in the next paragraph. At this point, a quick and dirty way of estimating the values $\Omega_n(\alpha)$ for large n could be to take an exponential fit through some points already calculated for low n and having the same limiting value for $n \to \infty$. However, from a theoretical point of view, this is highly unsatisfactory. We have made considerable effort to devise an algorithm that produces the quantities $\Omega_n(\alpha)$ in a direct way based on (329), but did not succeed. Nevertheless, these quantities can be obtained numerically using an entirely different approach, as we will see further.

The case $N \to \infty$: formation of a barrier and decoupling

If N is chosen larger, the packets of the same type will more and more cluster together in the queue. As illustrated in Fig. 4.15, a typical pattern will gradually emerge. In this figure, the content of the system is shown during slot 561 of a simulation with $N = 5, 10, 15, 20$ and with the arrival process chosen as in (282) where $\lambda_1 = \lambda_2 = 0.485$. One observes that at the front of the queue, there are mainly 1-packets with an occasional 2-packet in between. Behind the 1-packets, there is a contiguous pool of reservations that grows as N increases and after that, a swarm of 2-packets interspersed with reservations made by the 1-packets.

It is this gradually emerging contiguous 'pool' of reservations immediately behind the 1-packets in the queue that plays a crucial role. This group of reservations forms as it were a *barrier* for the 2-packets, which increases in size as N gets larger and is therefore harder and harder to penetrate. Let us now exactly define the barrier as the

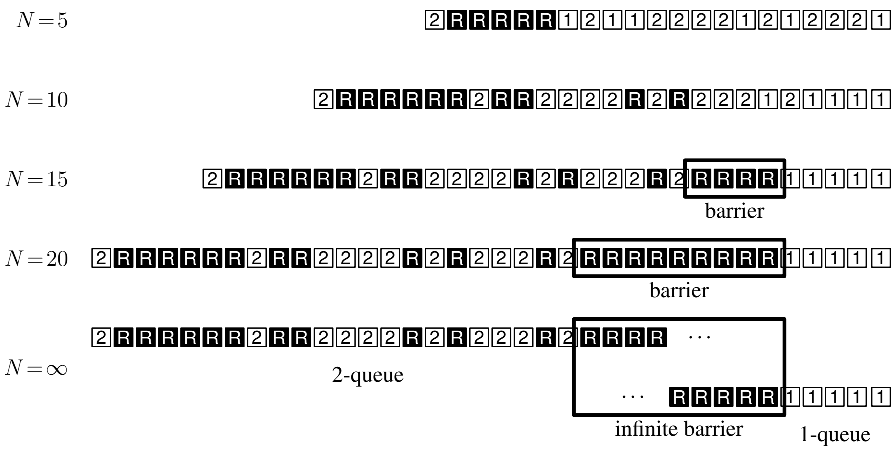


Figure 4.15: When N gets larger, a barrier forms between the 1- and 2-packets. For $N = \infty$ the system decouples into two logical sub-queues of which both the behaviour and function correspond to the low- and high-priority queues in an AP system.

contiguous row of reserved spaces that is positioned *behind* the last 1-packet, but in *front* of the first 2-packet in the queue. As such, it happens that for low N, this barrier does not exist, as is e.g. the case in Fig. 4.15 for $N = 5, 10$.

There are now two antagonistic effects that can cause the barrier to grow and decline respectively, during the evolution of the system. These effects can be understood in a qualitative way as follows. Recall that when no 1-packets are present in the queue, a 2-packet will *jump* to the server if there are only reservations on the positions between itself and the server. If there are many reservations, it is clear that this is the usual way in which the 2-packets gain access to the server. Every time such a jump takes place, a 2-packet disappears from the swarm at the end of the queue. Additionally, if one or more reservations were positioned behind this 2-packet, these will become part of the barrier after the jump. In this way, we see that whenever a 2-packet jumps, the barrier may grow with one or more units. Keep in mind however, that this can only occur if there are no 1-packets in the system. In contrast, the barrier will decrease in size every time a new 1-packet is stored in the queue. This packet will take the place of the first reservation in the queue, which is also the first reservation in the barrier, at least if there *is* a barrier at that moment. Also, the *new* reservation that is made by the 1-packet is placed at the end of the queue and will eventually contribute to the dispersed character of the swarm with 2-packets there. There is also an intermediate situation, in which the barrier neither decreases nor increases during a certain slot. This is e.g. the case if there are 1-packets present in the system, but no new 1-packets arrive.

It is now clear which conditions are responsible for the formation of a large barrier between the (swarm of) 2-packets at the end of the queue and the 1-packets at the front. Obviously, the number of reservations N in the system is important: the barrier can never grow beyond N positions. Note that this number N is also the initial size of the barrier if we start from an empty system. If the system is empty, all reservations are placed on the first N positions of the queue, which forms already a maximal barrier if the first packets are stored. From then on, it is the balance between the two mentioned antagonistic effects that determines the further evolution of the barrier's size. So now, from the discussion above, we can conclude that the growth of the barrier is stimulated by the sufficiently long *absence of 1-packets* from the system. This is for instance the case for small λ_1/λ_T, i.e. if the fraction of 1-packets in the traffic is small. In those circumstances, the growing effect because of the 2-packets jumping to the server is stronger than the decline effect because of 1-packets being stored in the queue.

Once there is a substantial barrier, the 2-packets can only reach the server by *jumping* over the reservations in the barrier. In contrast, we have seen that a 2-packet can also reach the server by shifting towards lower positions in a natural way and thus reaching the group of 1-packets at the front of the queue. But before this can happen the barrier must first have disappeared completely, which is exactly what we can see as a 2-packet breaching through, or penetrating the barrier. Note that a 2-packet can only breach the barrier if during some period enough 1-packets arrive in the system. On the other hand, under circumstances in which the growing of the barrier is stimulated (high N and few arriving 1-packets), fewer and fewer 2-packets will be able to breach through and most 2-packets will have to jump to reach the server.

Eventually, if $N = \infty$, the queue will decouple into two logical sub-queues: one for the 1-packets at the front closest to the server and one containing a swarm of 2-packets at the far end. We call these sub-queues the 1-queue and the 2-queue respectively. In between these sub-queues there is an impenetrable barrier containing an infinite number of reservations (see Fig. 4.15) that causes the decoupled operation of the system. No matter how many 1-packets arrive and use up reservations from the barrier, the barrier stays infinitely large and no 2-packet can ever breach through. Consequently, the 2-packets can only reach the server by jumping in this situation.

It is seen that for $N = \infty$, the packets in the reservation system behave the same as in the AP system. Arrivals of type 1 will always be stored in the first reservation of the barrier and thus find connection to the logical 1-queue. On the other hand, the reservation protocol dictates that the arriving 2-packets are stored at the end of the queue and therefore become part of the logical 2-queue. If the server becomes available, the next packet that is scheduled for service is the one positioned closest to the server. Since the 1-packets are grouped on the first positions, the server always schedules a 1-packet if one is available. Only if there are no 1-packets present, a 2-packet will jump over the (infinite) barrier to be served next. From the numerical examples further on, we will indeed observe that for high N the studied delay performance converges towards that of the AP system.

The decoupled 1-queue

We can use this resemblance of the logical 1-queue in a ∞R-system to the high-priority queue under AP, to obtain the limiting distribution $\Omega_\infty(z)$ of the position $\hat{m}_1^{[\infty]}$ of the first reservation. Let u_1 be the number of packets in the high-priority queue in an AP system subjected to the same arrival process as we consider here. Since these high-priority packets are not in any way affected by the low-priority packets (see [196]), it is clear that this high-priority queue behaves as a GI-1-1 queue with the number of arrivals per slot distributed as $A_1(z)$. Hence, similar to (251), we have for the pgf $U_1(z)$ of u_1:

$$U_1(z) \;=\; (1-\lambda_1)\,\frac{A_1(z)(z-1)}{z-A_1(z)}\,. \tag{330}$$

Now, assuming that u_1 is also the number of 1-packets in the logical 1-queue, we have

$$\hat{m}_1^{[\infty]} \;=\; m_1^{[\infty]}-1=(u_1-1)^+\,. \tag{331}$$

With (330), it then easily follows that

$$\Omega_\infty(z) \;=\; (1-\lambda_1)\frac{z-1}{z-A_1(z)}\,, \qquad \text{and} \quad \Omega_\infty(\alpha) \;=\; (1-\lambda_1)\frac{\alpha-1}{\alpha-A_1(\alpha)}\,, \tag{332}$$

where the latter is the limiting value in Fig. 4.14. In case of AP (or $N=\infty$), the number of packets that will be served no later than $\mathcal{P}$ is $(u_1-1)^+ + \ell_1$ which is also the delay of $\mathcal{P}$. Hence, the pgf of the delay is given as the product of (319) and (322) which gives (see also [196])

$$D_1^{\mathrm{AP}}(z) = \frac{1-\lambda_1}{\lambda_1}\,z\,\frac{A_1(z)-1}{z-A_1(z)}\,. \tag{333}$$

A generating function of generating functions

Whereas calculating the functions $\Omega_n(z)$, $n=1,2,\dots$ iteratively from (329) is difficult, taking the transform of this sequence is much easier. Specifically, let us define

$$\Omega(x,z) \triangleq \sum_{n=1}^{+\infty}\Omega_n(z)x^n\,, \tag{334}$$

which is the generating function of the generating functions $\Omega_n(z)$. From (329) we find a nice closed-form expression for $\Omega(x,z)$:

$$\Omega(x,z) \;=\; \frac{p_0(z-1)\,\Phi(x,z) - \frac{z-1}{\alpha-1}(A_1(\alpha x)-\alpha)\,\Omega(x,\alpha)}{z-A_1(zx)}\,, \tag{335}$$

where we have used the x-transform of the functions $\Phi_n(z)$ as well:

$$\Phi(x,z) \triangleq \sum_{n=1}^{+\infty}\Phi_n(z)x^n\,. \tag{336}$$

From (326) we find for $\Phi(x,z)$:

$$\Phi(x,z) \;=\; \frac{x}{1-x}\Big[\frac{A_1(z)-A_1(zx)}{z-A_T(z)} - \frac{A_1(\alpha)-A_1(\alpha x)}{\alpha-A_T(\alpha)}\Big]\,. \tag{337}$$

This approach has been used before, e.g. [40,56,198], in the context of transient analysis of queues. In those papers, the authors look at the queue content distribution of a particular system in subsequent slots $1, 2, 3, \ldots$ assuming a certain system state in slot 0. As we do, they work with a double transform, one with respect to the queue content and one with respect to the discrete time parameter. Evidently, the difference with our situation here is that in (334), the z-transform is with regard to the position of the first reservation position while the x-transform corresponds to the number of reservations in the system.

Now we are faced with the same kind of problem as before. How do we determine $\Omega(x,\alpha)$, the x-transform of the elusive sequence $\Omega_n(\alpha)$, $n \geqslant 1$? Taking $z \to 0$ in (335) yields

$$\Omega(x,0) \;=\; \frac{p_0}{\alpha}\,\frac{x}{x-1}\,\frac{A_1(\alpha)-A_1(\alpha x)}{\alpha - A_T(\alpha)} \;+\; \frac{A_1(\alpha x)-\alpha}{\alpha(1-\alpha)}\,\Omega(x,\alpha)\,, \tag{338}$$

which is of no help, since it only determines $\Omega(x,\alpha)$ as a function of the equally unknown $\Omega(x,0)$. Note that (338) is in fact the transform of relation (327). A better idea is the following. The function $\Omega(x,z)$, being the transform of probability generating functions, is known to be analytic for x and z lying in the unit disc. If we could find a pair (x,z) in that region for which the denominator in (335) becomes zero, then we know the numerator should be zero as well. Fortunately, one can invoke Rouché's theorem (see p. 159) to show that if $|x|<1$, there always exists a unique $\hat{Y}(x)$ for which $|\hat{Y}(x)|<1$ and that satisfies

$$\hat{Y}(x) \;=\; A_1\bigl(x\,\hat{Y}(x)\bigr)\,, \qquad \text{with} \quad \hat{Y}(1)=1\,. \tag{339}$$

Hence, if we let $z \to \hat{Y}(x)$ in (335), the numerator must vanish, which yields

$$\Omega(x,\alpha) \;=\; p_0(\alpha-1)\,\frac{\Phi(x,\hat{Y}(x))}{A_1(\alpha x)-\alpha}\,. \tag{340}$$

This, together with (337), allows us to write (335) as

$$\Omega(x,z) \;=\; p_0\frac{x}{1-x}\,\frac{z-1}{z-A_1(zx)}\left[\frac{A_1(z)-A_1(zx)}{z-A_T(z)} - \frac{A_1(\hat{Y}(x))-\hat{Y}(x)}{\hat{Y}(x)-A_T(\hat{Y}(x))}\right]. \tag{341}$$

This result determines the sequence of pgfs $\Omega_1(z), \Omega_2(z), \ldots$, due to definition (328). Let us assume a fixed (complex) value of z, then it is possible to obtain $\Omega_n(z)$ by inverting the x-transform (341). There exist many numerical methods to obtain the coefficients $[x^n]\Omega(x,z)$ of a generating function and most of them involve the evaluation of $\Omega(x,z)$ on a number of discrete points on a contour C around the origin in the x-plane. For instance, the inversion method in [20] which we also used in Sec. 3.8, uses a circular contour C_r of radius $0<r<1$ around $x=0$. However, the problem with the evaluation of $\Omega(x,z)$ now is that the function $\hat{Y}(x)$ appearing in (341) is *not known explicitly*. Indeed, we know that $\hat{Y}(x)$ exists and is unique, but we only have the implicit relation (339) to determine it. This complicates matters a bit, since every time we want to evaluate $\Omega(x,z)$ for a certain x on C_r (and a certain z, of course), we also have to determine $\hat{Y}(x)$ numerically from (339). To find this value, one can choose any complex root-finding algorithm to find the root $z^* = \hat{Y}(x)$ of $z^* - A_1(z^*x)$.

We can apply this numerical inversion method particularly in case $z=\alpha$, i.e. to obtain the quantities $\Omega_n(\alpha)$, $n=1,2,\ldots$ which we will need to obtain the mean value of the type 1 packet delay. Although this method is computationally rather heavy, (partly due to having to solve for the root $\hat{Y}(x)$ anew each time), it allows to obtain accurate values of $\Omega_n(\alpha)$ within reasonable time for far greater n than in Fig. 4.14. Plotting out these values confirms that the sequence converges to a limiting value $\Omega_\infty(\alpha)$. Obviously, we expect the sequence of pgfs $\Omega_n(z)$ to converge to a limiting distribution $\Omega_\infty(z)$ as well.

Without much additional effort, it is also possible to obtain this limiting pgf $\Omega_\infty(z)$ directly from (341) as follows. In general, suppose we have sequence f_n, $n\geqslant 1$ that converges to some (unknown) limiting value $f_\infty>0$, or in other words, $f_\infty=\lim_{n\to\infty} f_n$ exists. For the generating function $F(x)=f_1x+f_2x^2+\ldots$ we then have that $F(1)=\infty$. Let us split up this generating function in two terms $F(x)=F^*(x)+F_c(x)$ such that

$$F^*(1)<\infty\,,\qquad \text{and}\qquad F_c(1)=\infty\,, \tag{342}$$

i.e. a power series that converges in $z=1$ and one that does not. The coefficients $f_n^*=[x^n]F^*(x)$ then must form a sequence that converges to 0 (see [99], p. 187), and consequently, the coefficients $[x^n]F_c(x)$ must converge to f_∞, just like the original sequence f_n. Suppose we choose for $F_c(x)$:

$$F_c(x)\;=\;f_\infty x+f_\infty x^2+f_\infty x^3+\ldots\;=\;f_\infty\,\frac{x}{1-x}\,,$$

where obviously $|x|<1$. For $x=1$, this series diverges as required by (342). Choosing $F_c(x)$ this way implies that $F^*(x)$ is the generating function of the sequence $f_n^*=f_n-f_\infty$:

$$F^*(x)\;=\;(f_1-f_\infty)x+(f_2-f_\infty)x^2+(f_3-f_\infty)x^3+\ldots\;=\;F(x)-f_\infty\frac{x}{1-x}\,. \tag{343}$$

The limiting value f_∞ can now be found by insisting that (342) holds, i.e. taking $x\to 1$ in (343) must yield a finite number. Applied to the sequence $\Omega_n(z)$, we can thus find the limiting distribution $\Omega_\infty(z)$ by requiring that

$$\lim_{x\to 1}\,\Omega(x,z)-\Omega_\infty(z)\,\frac{x}{1-x}\;=\;\lim_{x\to 1}\,\frac{(1-x)\,\Omega(x,z)-x\,\Omega_\infty(z)}{1-x}<\infty\,. \tag{344}$$

As the denominator becomes 0 for $x\to 1$, the numerator must be zero as well, which together with (341) results in the relation

$$\Omega_\infty(z)\;=\;\lim_{x\to 1}\frac{p_0(z-1)}{z-A_1(zx)}\left[\frac{A_1(z)-A_1(zx)}{z-A_T(z)}-\frac{A_1(\hat{Y}(x))-\hat{Y}(x)}{\hat{Y(x)}-A_T(\hat{Y}(x))}\right]. \tag{345}$$

In the limit, the first term vanishes while the second one must be evaluated using de l'Hôpital's rule, which requires the value of $\hat{Y}'(1)$. Note that from (339), we can derive that

$$\hat{Y}'(1)\;=\;\frac{\lambda_1}{1-\lambda_1}\,. \tag{346}$$

Taking the limit in (345), we find exactly (332) for the expressions of $\Omega_\infty(z)$ and $\Omega_\infty(\alpha)$. Apparently, we obtain exactly the same result here by using $\Omega(x,z)$ indigenous to the reservation discipline, as we did in the previous paragraph by assuming a decoupled queue with the logical 1-queue behaving as the high-priority queue under AP.

Mean value of the type 1 packet delay

For the pgf $D_1(z)$ of the type 1 packet delay, we now have from (321)–(323) and definition (328) that

$$D_1(z) \;=\; \sum_{j=1}^{N}\omega_j z^j \Omega_{N-j+1}(z) + p_0\,\frac{z-1}{z-A_T(z)}\Big(L_1(z)-\sum_{j=1}^{N}\omega_j z^j\Big)\,, \tag{347}$$

where the functions $\Omega_n(z)$ follow from the discussion in the previous paragraph. As such, $D_1(z)$ can be evaluated numerically for any particular z. On the other hand, if N is not too large, one can endeavour to obtain the functions $\Omega_n(z)$ analytically from the recursive relation (329) as well. Invoking the moment-generating property of pgfs, the expected value $\mathrm{E}[d_1]$ of the delay experienced by an arbitrary 1-packet $\mathcal{P}$ can be obtained as the first derivative of $D_1(z)$ evaluated in $z=1$. Using the notations in (238)–(241) and (319) gives

$$\mathrm{E}[d_1] = D_1'(1) = \sum_{j=1}^{N}\omega_j\,\Omega'_{N-j+1}(1) + L_1'(1) + \frac{\lambda_T'}{2(1-\lambda_T)}\sum_{j=N+1}^{+\infty}\omega_j\,, \tag{348}$$

where $\Omega_n'(1)=\mathrm{E}[\hat{m}_1^{[n]}]$, $n=1,\ldots,N$ and where we have used the fact that $p_0=1-\lambda_T$.

Clearly, the problem at hand is now to determine the mean value $\mathrm{E}[\hat{m}_1^{[n]}]$ of the first reservation position in a system with n reservations, $n=1,\ldots,N$. In order to do so, we assume that the quantities $\Omega_n(\alpha)$ are available. As we discussed, they can either be obtained analytically by iterating (329) and taking the limit $z\to\alpha$ in each step, or they follow from the numerical inversion method discussed in the previous paragraph. Differentiating (329) to z and taking the limit $z\to 1$ on both sides, we find after some straightforward manipulations and using (326):

$$\begin{aligned}\mathrm{E}[\hat{m}_1^{[N]}] \;=\; &\frac{1}{1-\alpha}\Big[\lambda_1-1+\frac{\lambda_T'}{2(1-\lambda_T)}\sum_{i=N}^{+\infty}\beta_i-\frac{p_0}{\alpha-A_T(\alpha)}\sum_{i=N}^{+\infty}\beta_i\alpha^i\\ &+\frac{1}{1-\alpha}\sum_{i=1}^{N-1}\beta_i\alpha^i\Omega_{N-i}(\alpha)\Big]+\sum_{i=1}^{N-1}\frac{\beta_i}{1-\alpha}\mathrm{E}[\hat{m}_1^{[N-i]}]\,.\end{aligned} \tag{349}$$

As was the case with (329), this relation can be solved iteratively as well. The first iteration for $N=1$ yields $\mathrm{E}[\hat{m}_1^{[1]}]$, the second $\mathrm{E}[\hat{m}_1^{[2]}]$ for $N=2$ and so on. However, still assuming that we know the sequence $\Omega_n(\alpha)$, it is possible to provide a *direct* solution of the expected values $\mathrm{E}[\hat{m}_1^{[n]}]$ from (349). For the sake of transparency, let us define the following shorthands

$$\mu_n \triangleq \mathrm{E}[\hat{m}_1^{[n]}] \;=\; \Omega_n'(1)\,,\qquad n=1,\ldots,N\,, \tag{350}$$

$$\delta_i \triangleq \frac{\beta_i}{1-\alpha}\,, \qquad\qquad i>0\,, \tag{351}$$

$$\Gamma_n \triangleq \frac{1}{1-\alpha}\Big[\lambda_1-1+\frac{\lambda_T'}{2(1-\lambda_T)}\sum_{i=n}^{+\infty}\beta_i-\frac{p_0}{\alpha-A_T(\alpha)}\sum_{i=n}^{+\infty}\beta_i\,\alpha^i + \frac{1}{1-\alpha}\sum_{i=1}^{n-1}\beta_i\,\alpha^i\,\Omega_{n-i}(\alpha)\Big]\,. \tag{352}$$

This reduces (349) to

$$\mu_N \;=\; \Gamma_N+\sum_{i=1}^{N-1}\delta_i\,\mu_{N-i}\,. \tag{353}$$

In this relation, the quantities Γ_n, $n=1,\ldots,N$ and δ_i, $i\geqslant 1$ are fully known whereas the quantities μ_n are the mean values we seek. One can already deduce from (353) that each μ_n will be a linear combination of the quantities Γ_1 up to Γ_n with coefficients being a function of δ_1 up to δ_n. In order to find these coefficients we proceed as follows. Let us first arrange the values obtained from (352) in a $N\times 1$ matrix $\mathbf{\Gamma}$,

$$\mathbf{\Gamma}^T \triangleq \begin{bmatrix}\Gamma_1 & \Gamma_2 & \Gamma_3 & \ldots & \Gamma_N\end{bmatrix}\,. \tag{354}$$

Secondly, we use the values (351) to define the following $N\times N$ matrix,

$$\mathbf{H} \triangleq \begin{bmatrix} \delta_1 & 1 & 0 & 0 & \cdots & 0\\ \delta_2 & 0 & 1 & 0 & \cdots & 0\\ \delta_3 & 0 & 0 & 1 & \cdots & 0\\ \vdots & & & & \ddots & \\ \delta_{N-1} & 0 & 0 & 0 & \cdots & 1\\ \delta_N & 0 & 0 & 0 & \cdots & 0 \end{bmatrix}\,. \tag{355}$$

This matrix $\mathbf{H}$ is an instance of what is known as a *Leslie matrix* [125] due to P.H. Leslie who used this kind of matrices in 1945 for the study of population growth. In addition, let $\mathbf{e}$ be a row matrix of appropriate size with 1 as its first entry and all other elements 0, i.e.

$$\mathbf{e} \triangleq \begin{bmatrix}1 & 0 & 0 & \ldots & \end{bmatrix}\,. \tag{356}$$

In the current context, $\mathbf{e}$ is a $1\times N$ row matrix. Now, it can be verified that μ_n is obtained by calculating the $(n-1)$th power of $\mathbf{H}$, i.e. the solution of (353) is

$$\mu_n \;=\; \mathbf{e}\,\mathbf{H}^{n-1}\,\mathbf{\Gamma}\,, \qquad n=1,\ldots,N\,. \tag{357}$$

This provides the mean values $\mu_n=\Omega_n'(1)$ in the expression for the mean delay (348) which now becomes

$$\mathrm{E}[d_1^{[N]}] = \mathbf{e}\Big(\sum_{j=1}^{N}\omega_j\,\mathbf{H}^{N-j}\Big)\mathbf{\Gamma}+\frac{\lambda_T'}{2(1-\lambda_T)}\big(1-\sum_{j=1}^{N}\omega_j\big)+1+\frac{\lambda_1'}{2\lambda_1}\,, \tag{358}$$

where we have used (319) as well.

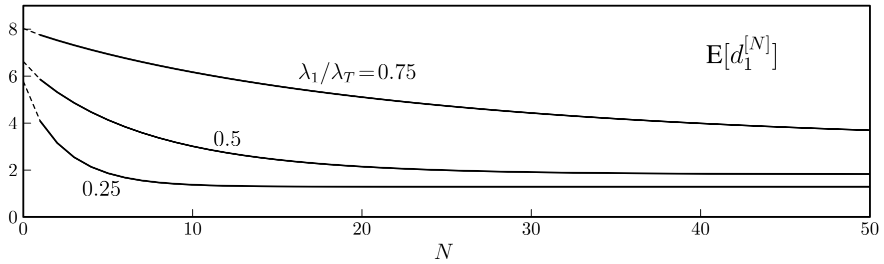


Figure 4.16: Mean value $E[d_1^{[N]}]$ of the type 1 packet versus the number of reserved spaces N, in case of arrival process (282), a total load of $\lambda_T=0.9$ and traffic mix $\lambda_1/\lambda_T=0.25, 0.5, 0.75$. The value for $N=0$ indicates the result for FIFO.

This latest expression allows us to calculate the mean delay of type 1 in the system with N reservations by means of $N-1$ matrix multiplications. However, in doing so, it is possible to arrange the calculations in such a way that the mean values $E[d_1^{[n]}]$ of the delay in the corresponding systems with *less* than N reservations are produced as well. In other words, calculating the delay in a system with one additional reservation requires only one additional matrix multiplication. The following algorithm shows how this can be achieved.

- For $n=1,\ldots,N$, calculate the values $\Omega_n(\alpha)$, either analytically or numerically, as explained before. Note that for high n, one could consider approximating $\Omega_n(\alpha)$ by the limiting value $\Omega_\infty(\alpha)$ given in (332).
- For $n=1,\ldots,N$, calculate the entries Γ_n in the matrix $\mathbf{\Gamma}$ using (352).
- Now construct the matrix $\mathbf{H}$ as in (355) and define the starting values

$$\psi_0 = \frac{\lambda_T'}{2(1-\lambda_T)} + 1 + \frac{\lambda_1'}{2\lambda_1} \qquad \text{and} \qquad \mathbf{Q}_0 = \mathbf{0}\,. \tag{359}$$

Then, for $n=1,\ldots,N$, calculate

$$\psi_n = \psi_{n-1} - \frac{\lambda_T'}{2(1-\lambda_T)}\,\omega_n\,, \qquad \text{and} \qquad \mathbf{Q}_n = \mathbf{Q}_{n-1}\,\mathbf{H} + \omega_n\mathbf{I}\,, \tag{360}$$

where $\mathbf{I}$ is the $N\times N$ identity matrix. As the mean value of $d_1^{[n]}$ follows from (358) for $N=n$, we can now see that after the nth step in this iteration, the mean delay of type 1 in a system with n reservations is given by

$$E[d_1^{[n]}] = \psi_n + \mathbf{e}\,\mathbf{Q}_n\,\mathbf{\Gamma}\,. \tag{361}$$

We have used this procedure to plot in Fig. 4.16 the mean delay versus the number of reserved spaces in the system. The arrival process was again taken to be the one in (282) with total load $\lambda_T=0.9$ and the maximal N considered is 50. The traffic mix λ_1/λ_T was subsequently chosen to be 0.25, 0.5 and 0.75. For each of the three curves, the values $\Omega_n(\alpha)$, $n=1,\ldots,50$ were determined first by numerical inversion of (341) for $z=\alpha$, with an accuracy of 6 digits. Then, the matrix multiplications in (361) were performed. On a 2800Mhz computer, each curve took about three minutes to compute.

Tail distribution of the type 1 packet delay

Another important characteristic of the delay distribution besides the mean value, is its tail distribution. As we did on p. 154 for the 1R-system, we can now also use the *dominant pole approximation* to derive the tail behaviour of the type 1 delay in case of multiple reservations. This approach asserts that the probability $\mathrm{Prob}[d_1^{[N]}=n]$ for large n can be approximated very accurately as

$$\mathrm{Prob}[d_1^{[N]}=n] \;\cong\; -\theta_1^{[N]}\, z_d^{-n-1}\,, \tag{362}$$

where z_d is the pole of $D_1(z)$ with smallest modulus and $\theta_1^{[N]}$ is the residue in z_d:

$$\theta_1^{[N]} = \mathrm{Res}_{z_d} D_1(z) = \lim_{z\to z_d} (z-z_d)D_1(z)\,. \tag{363}$$

Additionally, in order to assure a nonnegative mass function of the delay distribution we know that the dominant pole z_d must be real and positive. For further details, see also Sec. 3.B and [43, 46].

The first thing to do therefore, is to identify the dominant pole z_d of $D_1(z)$. After careful inspection of the expression (347), one can prove that its dominant pole can only originate from the factor $(z - A_T(z))^{-1}$ appearing in the second term, but also present in each $\Omega_n(z)$ through (326) and (329). Note that the multiplicity of this factor is equal to 1 in all of these terms. As such, z_d can be obtained numerically as the smallest real root larger than 1 of

$$z - A_T(z) = 0\,. \tag{364}$$

Note that this value is *independent* of N and identical to the dominant pole we had in case of FIFO and the 1R-system considered in Sec. 4.1. This is exactly what we expect to find based on the simulations in Fig. 4.7. Indeed, we see from (362) that on a logarithmic plot the slope of the tail distribution is given by the geometric decay rate z_d^{-1}. Looking at the distributions obtained by simulation, we observe for large n that the slope of the tail distribution is the same as in case of FIFO, no matter how many reservations there are in the system.

Secondly, we have to evaluate the limit in (363) to obtain the residue $\theta_1^{[N]}$. Fortunately, not *all* terms in $D_1(z)$ as given in (347) have z_d as a pole. Consequently, all the contributions to $D_1(z)$ that do not, will vanish when taking the limit $z \to z_d$, due to the factor $(z-z_d)$. This observation brings up the following idea. We have argued that $D_1(z)$, and more specifically the functions $\Omega_n(z)$, $n=1,\ldots,N$ which are foremost required, can in practice only be obtained analytically in case there are but very few reservations in the system. However, chances are that this task is much less complicated if we only consider the *contributions that have a pole in* z_d. In other words, we hope that the recursion (329) becomes easier to solve if we can neglect the terms that would vanish under the limit (363) anyway. Let us define $\Omega_n^*(z)$, $n=1,\ldots,N$ as these 'modified' versions of the original functions $\Omega_n(z)$ determined by (329), such that

$$\mathrm{Res}_{z_d}\Omega_n^*(z) \;=\; \lim_{z\to z_d}(z-z_d)\Omega_n^*(z) \;=\; \lim_{z\to z_d}(z-z_d)\Omega_n(z) \;=\; \mathrm{Res}_{z_d}\Omega_n(z)\,. \quad (365)$$

Note that these modified functions are no longer pgfs. Their only correct interpretation is having the same residue in z_d as the original pgfs. Removing all terms that do not have a pole in z_d, the modified version of (329) is now

$$\Omega_N^*(z) \;=\; p_0(z-1)\frac{f_N(z)}{z-\alpha} + \sum_{i=1}^{N}\beta_i\frac{z^i}{z-\alpha}\,\Omega_{N-i}^*(z)\,, \quad (366)$$

where we recall that $f_n(z)$ is defined in (325). The required residues (365) are therefore determined by the recursion

$$\mathrm{Res}_{z_d}\Omega_N(z) \;=\; \frac{p_0}{1-A_T'(z_d)}\,\frac{z_d-1}{z_d-\alpha}\sum_{i=N}^{+\infty}\beta_i z_d^i + \sum_{i=1}^{N}\beta_i\frac{z_d^i}{z_d-\alpha}\,\mathrm{Res}_{z_d}\Omega_{N-i}(z)\,, \quad (367)$$

where we have used de l'Hôpital's rule and definition (325). This relation can be represented much simpler if we introduce

$$\mu_n^* \triangleq \mathrm{Res}_{z_d}\Omega_n(z)\,, \qquad n=1,\dots,N\,, \quad (368)$$

$$\delta_i^* \triangleq \frac{\beta_i}{z_d-\alpha}\,z_d^i\,, \qquad i>0\,, \quad (369)$$

$$\Gamma_n^* \triangleq \frac{z_d-1}{z_d-\alpha}\,\frac{p_0}{1-A_T'(z_d)}\sum_{i=n}^{+\infty}\beta_i\,z_d^i\,, \qquad n=1,\dots,N\,, \quad (370)$$

similar to (350)–(352). Relation (367) then becomes

$$\mu_N^* \;=\; \Gamma_N^* + \sum_{i=1}^{N-1}\delta_i^*\,\mu_{N-i}^*\,, \quad (371)$$

which is symbolically *exactly* the same as (353) and therefore has the same kind of solution:

$$\mu_n^* \;=\; \mathrm{Res}_{z_d}\Omega_n(z) \;=\; \mathbf{e}\,(\mathbf{H}^*)^{n-1}\,\boldsymbol{\Gamma}^*\,, \qquad n=1,\dots,N\,. \quad (372)$$

Here, the matrix $\mathbf{H}^*$ is the same as $\mathbf{H}$, but with every entry δ_i replaced by δ_i^*. Using this solution, we finally find from (347) for the residue $\theta_1^{[N]}$:

$$\theta_1^{[N]} = \mathbf{e}\left(\sum_{j=1}^{N}\omega_j\,z_d^j\,(\mathbf{H}^*)^{N-j}\right)\boldsymbol{\Gamma}^* + p_0\frac{z_d}{\lambda_1}\,\frac{A_1(z_d)-1}{1-A_T'(z_d)} + p_0\frac{1-z_d}{1-A_T'(z_d)}\sum_{j=1}^{N}\omega_j\,z_d^j\,, \quad (373)$$

where we have used expression (319) for $L_1(z)$.

As with the mean value of the delay, the calculation of the residue $\theta_1^{[N]}$ can be performed in such a way that the equivalent residues $\theta_1^{[n]}$ for systems with fewer than N reservations are produced as well. The following algorithm implements this.

- ► For $n=1,\ldots,N$, calculate the entries Γ_n^* in the matrix $\mathbf{\Gamma}^*$ using (370).
- ► Now determine the values δ_i^* as in (369) and populate the matrix $\mathbf{H}^*$. The residues $\theta_1^{[1]}$ to $\theta_1^{[N]}$ can now progressively be obtained as follows. Define the starting values

$$\psi_0^* = p_0 \, \frac{z_d}{\lambda_1} \, \frac{A_1(z_d)-1}{1-A_T'(z_d)} \,, \qquad \text{and} \qquad \mathbf{Q}_0^* = \mathbf{0} \,. \tag{374}$$

Then, for $n=1,\ldots,N$, calculate

$$\begin{aligned} \psi_n^* &= \psi_{n-1}^* + p_0 \frac{1-z_d}{1-A_T'(z_d)} \, \omega_n \, z_d^n \,, \\ \mathbf{Q}_n^* &= \mathbf{Q}_{n-1}^* \, \mathbf{H}^* + \omega_n \, z_d^n \, \mathbf{I} \,. \end{aligned}$$

After each step, the residue $\theta_1^{[n]}$ then follows from (373) as

$$\theta_1^{[n]} \;=\; \psi_n^* + \mathbf{e} \, \mathbf{Q}_n^* \, \mathbf{\Gamma}^* \,. \tag{375}$$

We have checked that for $N=1$, this algorithm produces exactly the same expression as (280).

4.3.5 Delay of type 2 packets

In this section, we turn our attention to the delay experienced by an arbitrary packet of type 2. Let us call $\mathcal{P}$ such an arbitrary packet and follow the events as it arrives in the system, advances through the queue and finally gets served and leaves. By giving a stochastic description of these events, we are able to obtain the distribution of the delay d_2 defined as the total number of slots that $\mathcal{P}$ is present in the system. As before, we refer to the slot during which $\mathcal{P}$ arrives as slot I. Recall also that in equilibrium, the system state distribution at the start of slot I is the same as in an arbitrarily chosen slot, due to the fact that the arrivals are independent from slot to slot. This means that we can call upon the results of Sec. 4.3.2 again to provide the joint distribution of the system state variables at the start of slot I. As in Sec. 4.1.3, let us denote the number of 1- and 2-arrivals in slot I as a_1^I and a_2^I respectively and let $\ell_2 \leqslant a_2^I$ indicate how many of these 2-arrivals are stored in the queue no later than $\mathcal{P}$. Expression (267) still applies for the joint pgf of a_1^I and ℓ_2.

Compared to the delay analysis of the 1-packets, an important complication is now that the delay of $\mathcal{P}$ not only depends on the system state and arrivals *during* slot I but also on the arrivals of type 1 occurring *after* slot I. In the case of a system with one reservation we have seen that the delay of $\mathcal{P}$ depends on whether or not the (single) R is seized by a 1-packet prior to the service of $\mathcal{P}$. If it is, this 1-packet moves into a position closer to the server than $\mathcal{P}$ and will therefore be served earlier than $\mathcal{P}$. As a consequence, the delay of $\mathcal{P}$ is extended by one slot compared to the case in which the R in front of $\mathcal{P}$ is *not* seized.

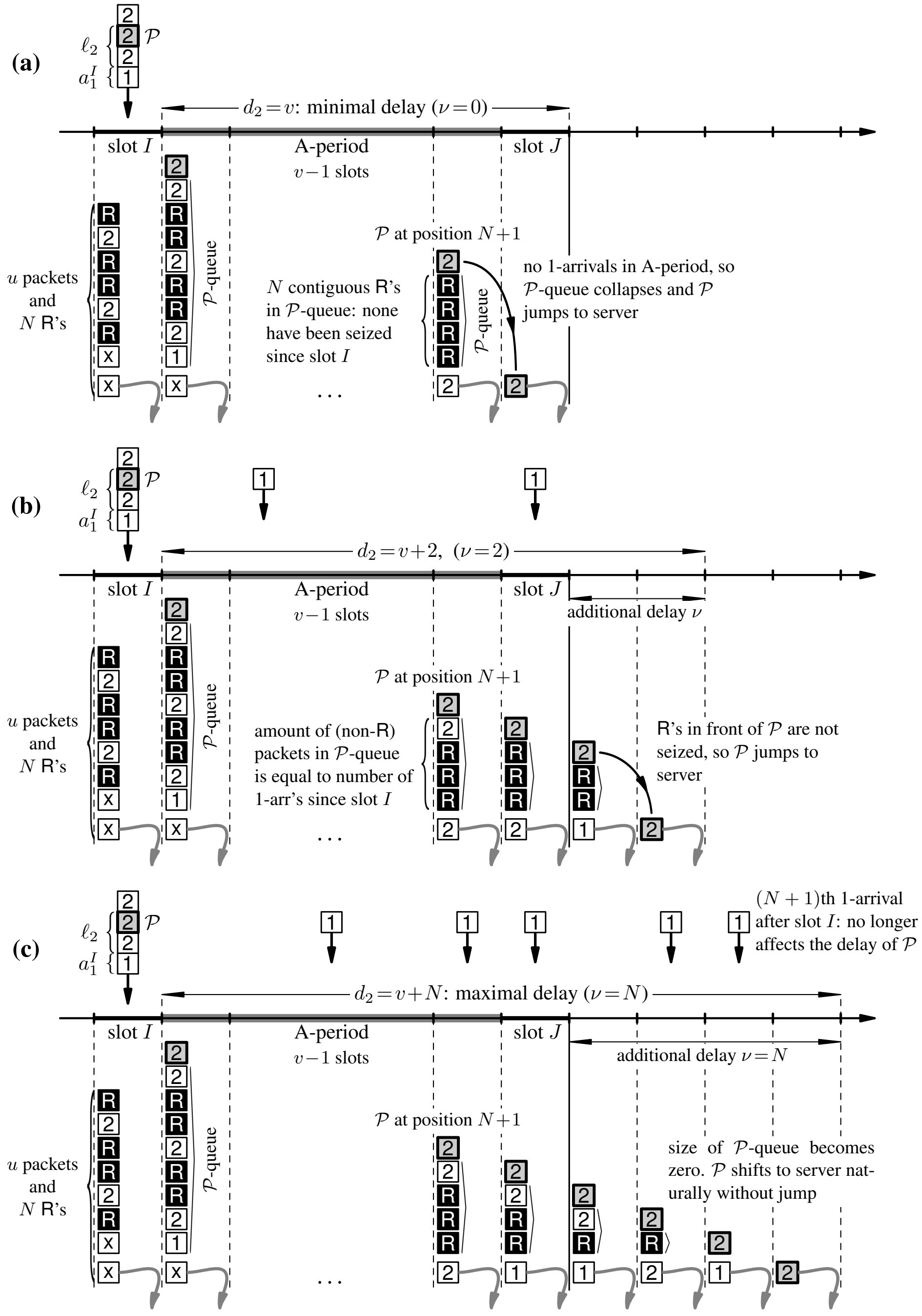


Figure 4.17: The delay of an arbitrary 2-packet $\mathcal{P}$ in a system with $N = 4$ reservations is influenced by 1-arrivals later than slot I. In **(a)** there are no 1-arrivals before the service of $\mathcal{P}$, so $\nu = 0$ and $d_2 = v$. In **(b)**, 2 reservations in front of $\mathcal{P}$ are seized extending the delay by two slots: $\nu = 2$. In **(c)**, there are so many 1-arrivals after slot I that all N reservations are seized, $\nu = N = 4$ and the delay has maximum value $d_2 = v + N$.

In case of a system with multiple reservations the same phenomenon occurs, only now there are N R's in front of $\mathcal{P}$ immediately after it is stored in the queue at the end of slot I. The delay d_2 depends on how many of these reservations will be seized during the time between slot I and the service of $\mathcal{P}$. This time is in fact the *waiting time* of $\mathcal{P}$. Let us denote by ν the number of reservations in front of $\mathcal{P}$ that will be seized. Clearly, these ν seized R's represent ν packets of type 1 belatedly positioned in front of $\mathcal{P}$. Hence, the delay of $\mathcal{P}$ will be extended by ν slots compared to the case in which *no* reservations are seized during the waiting time of $\mathcal{P}$, i.e. $\nu = 0$. For this reason, we refer to ν as the *additional delay*.

Although the mechanism that causes the delay to be extended by ν slots must certainly be accounted for, the delay of $\mathcal{P}$ is also contributed to by the packets already queued when $\mathcal{P}$ arrives. Suppose now that $\nu = 0$. In that case, all the packets that will be served during the delay of $\mathcal{P}$ are already present in the queue at the end of slot I. As in Sec. 4.1, let v indicate this amount of packets. Obviously, the delay of $\mathcal{P}$ expressed as a number of slots is then given by the variable v under these circumstances as well. Considering the content of the queue at the start of slot I and the arrivals during slot I, recall that v is to be expressed as

$$v \triangleq (u-1)^+ + a_1^I + \ell_2\,, \tag{376}$$

with u the content of the system. The corresponding pgf $V(z)$ is given in (268). If we also account for the additional delay ν, then we simply have that

$$d_2 = v + \nu\,. \tag{377}$$

It should be stressed again that ν only counts those reservations that

- were in existence at the end of slot I (after $\mathcal{P}$ is stored), therefore also positioned *in front of* $\mathcal{P}$, and
- are seized after slot I but before the service of $\mathcal{P}$.

Reservations *created* after slot I are of no concern to us since they are always positioned *behind* $\mathcal{P}$ and any 1-packet taking its position will not affect the delay of $\mathcal{P}$. As there are maximally N reservations, we thus have the bounds

$$0 \leqslant \nu \leqslant N\,. \tag{378}$$

Given a certain configuration of the queue at the end of slot I, the quantity v can be regarded as the *minimum* delay that $\mathcal{P}$ can have. On the other hand, if all N reservations positioned in front of $\mathcal{P}$ are seized during its waiting time, then $\mathcal{P}$ will experience a *maximal* delay of $v + N$ slots. From (377) and (378), we see that the delay is thus bounded as

$$v \leqslant d_2 = v + \nu \leqslant v + N\,. \tag{379}$$

To make things clear, we have given in Fig. 4.17 some examples that show how 1-arrivals after slot I affect the delay of $\mathcal{P}$. In the three cases **(a)**, **(b)** and **(c)**, the situation in slot I is identical: the same system state at the start of I and the same arrivals during slot I such that $\mathcal{P}$ is the second of an arriving batch of three 2-packets. After slot I however, the arrivals are different in each case. In **(a)**, there are no 1-arrivals during the waiting time of $\mathcal{P}$ so none of the R's in the queue are taken and $\nu = 0$. Given the situation in slot I, the resulting delay has the lowest possible value $d_2 = v$. In **(b)**, 2 of the 4 R's in front of $\mathcal{P}$ are seized before $\mathcal{P}$ is served, causing the delay to be extended by exactly 2 slots: $\nu = 2$. In **(c)**, *all* N available reservations in front of $\mathcal{P}$ are taken by 1-packets. Note that the last drawn 1-arrival is stored in time before $\mathcal{P}$'s waiting time ends, but does not cause the delay to be extended by another slot. The obvious reason is that all reserved positions in front of $\mathcal{P}$ are already occupied and the 1-packet ends up in the queue *behind* $\mathcal{P}$.

In our analysis, the following definitions prove to be useful. Let J be the slot in which $\mathcal{P}$ will be served if there is no additional delay ($\nu = 0$), then clearly

$$J = I + v \,. \tag{380}$$

Also, let us refer to the slots *between* slots I and J as the *A-period*. The length of this A-period is therefore equal to $v-1$ slots and has pgf $\hat{V}(z)$ given by

$$\hat{V}(z) \triangleq \frac{V(z)}{z} = \frac{p_0}{\lambda_2} \frac{A_T(z) - A_1(z)}{z - A_T(z)} \,, \tag{381}$$

which follows directly from (268). Both slot J and the A-period are indicated in the examples of Fig. 4.17. The importance of the A-period is the following. If *no* 1-arrivals occur during the A-period as in **(a)**, then there will be no additional delay either. Otherwise, the number of 1-arrivals during the A-period directly determines the distribution of ν as will be shown later.

Another useful concept to introduce is that of the $\mathcal{P}$-queue, indicated in Fig. 4.17 as well. As soon as $\mathcal{P}$ is stored in the queue, we refer to *the positions in front of* $\mathcal{P}$ as the $\mathcal{P}$-queue, *not* including the server at position 0. When $\mathcal{P}$ is stored in slot I, the $\mathcal{P}$-queue will contain exactly N reserved spaces, while the other positions of the $\mathcal{P}$-queue are occupied with packets of either type. As time progresses, both the size and the content of the $\mathcal{P}$-queue changes. Its size decreases with one position in every slot as the packets in the queue leave for service. The contents of the $\mathcal{P}$-queue changes as well, not only because packets leave but also because 1-arrivals may seize the reservations in the $\mathcal{P}$-queue as long as there are any available. The $\mathcal{P}$-queue ceases to exist once $\mathcal{P}$ enters service and its lifespan is therefore equal to the waiting time of $\mathcal{P}$. We say that the $\mathcal{P}$-queue *collapses* when $\mathcal{P}$ moves to the server. The exact delay of $\mathcal{P}$ can be inferred from the evolution of the $\mathcal{P}$-queue both in terms of its size and its contents, as we will make clear. It is important to observe the following fact concerning the $\mathcal{P}$-queue in the last slot of the A-period, i.e. slot $J-1$.

- ► During slot $J-1$, $\mathcal{P}$ is *always* in position $N+1$, regardless how many R's were seized between slot I and slot $J-1$. Consequently, the size of the $\mathcal{P}$-queue is N at that moment.

The collapse of the $\mathcal{P}$-queue can occur in two ways. As its size decreases by one position every slot, the size can eventually become zero, which means that $\mathcal{P}$ is in position 1 and will be served in the next slot. This is what happens in Fig. 4.17**(c)**. However, the waiting time of $\mathcal{P}$ can end *before* the size of the $\mathcal{P}$-queue ever reaches 0. Indeed, suppose that in one of the slots $J+k$, $k \geqslant 0$, the $\mathcal{P}$-queue contains some R's, but no real packets. If no 1-arrivals occur during the remainder of this slot to seize one of these reservations, then $\mathcal{P}$ will be the packet closest to the server in the next slot. Consequently, it will be chosen for service and *jump* over the R's in the $\mathcal{P}$-queue into the server.

The main part of our analysis will be devoted to finding the distribution of ν. An important thing to keep in mind is that v and ν are clearly *not* statistically independent. Indeed if v is high, this means that the A-period is long and we expect a lot of 1-arrivals in that period. As such, a lot of the R's in front of $\mathcal{P}$ will be seized by the time slot J starts, so ν will most likely be high as well. As a result, we expect a positive correlation between the quantities v and ν. In our analysis, we deal with this by conditioning on the value of v, i.e.

$$D_2(z) \;=\; \mathrm{E}[z^{d_2}] \;=\; \mathrm{E}[z^{v+\nu}] \;=\; \sum_{i=1}^{+\infty} \mathrm{E}[z^{i+\nu}\{v=i\}]\,. \tag{382}$$

Virtual content of the $\mathcal{P}$-queue: a PH-type distribution

We know that the $\mathcal{P}$-queue decreases in size by one position per slot. We also know for sure that immediately before slot J, the $\mathcal{P}$-queue is exactly N positions large. That moment, just before the end of the A-period when all arriving packets have been stored, is a decisive moment in the analysis of the delay. The contents of the $\mathcal{P}$-queue at that time, as well as arrivals of type 1 in the following slots $J, J+1, \ldots$ determine exactly when $\mathcal{P}$ will reach the server. As we have indicated before, the number of packets in the $\mathcal{P}$-queue at that moment is exactly given by the number of R's in the $\mathcal{P}$-queue that have been seized during the A-period.

For now, let us assume that the $\mathcal{P}$-queue is *not* limited in size, i.e. it can contain any number of packets and has an infinite number of reservations that can be seized. This abstraction admittedly seems to be strange at first, but is useful nonetheless. We will correct our results later on, to account for the actual size of the $\mathcal{P}$-queue. We refer to the number of packets in this fictitious $\mathcal{P}$-queue immediately *before* slot $J+k$ as the *virtual content* q_k. If for any $k=0,1,2,\ldots$ the virtual content q_k should become 0, then we know the $\mathcal{P}$-queue only contains reservations and no packets. As a consequence, $\mathcal{P}$ is the closest packet to the server then, and will be chosen for service in slot $J+k$. So what we want to find out is *when* the virtual content becomes zero and the (virtual) $\mathcal{P}$-queue collapses.

Our first task is to obtain the distribution of q_0, the virtual content before slot J starts. Under the assumption of an infinite $\mathcal{P}$-queue, the virtual content at this moment is exactly equal to the number of 1-arrivals during the A-period. Therefore, given that the A-period is i–1 slots long, i.e. $v=i$ $(i\geqslant 1)$, the mass function of q_0 is the $(i$–$1)$-fold convolution of the mass function of a_1, the number of 1-arrivals per slot. We denote

$$\zeta_{n,i} \triangleq \mathrm{Prob}[q_0=n|v=i]\,, \quad n\geqslant 0\,, \qquad \text{s.t.} \quad \sum_{n=0}^{+\infty} \zeta_{n,i}\, z^n = \big(A_1(z)\big)^{i-1}\,. \tag{383}$$

Let us arrange these probabilities in the matrix $\boldsymbol{\zeta}_i$ as

$$\boldsymbol{\zeta}_i \triangleq \begin{bmatrix}\zeta_{0,i} & \zeta_{1,i} & \zeta_{2,i} & \ldots\end{bmatrix}. \tag{384}$$

The probability matrix $\boldsymbol{\zeta}_i$ can be obtained from the mass function of the number of 1-arrivals a_1 per slot in (293) as

$$\boldsymbol{\zeta}_i = \mathbf{e}\,\mathbf{B}^{i-1}\,, \qquad i \geqslant 1\,, \tag{385}$$

where $\mathbf{e}$ is given by (356) and the matrix $\mathbf{B}$ is defined as the upper triangular Toeplitz matrix

$$\mathbf{B} \triangleq \begin{bmatrix} \alpha & \beta_1 & \beta_2 & \beta_3 & \cdots \\ 0 & \alpha & \beta_1 & \beta_2 & \ddots \\ 0 & 0 & \alpha & \beta_1 & \ddots \\ 0 & 0 & 0 & \alpha & \ddots \\ \vdots & \vdots & \vdots & & \ddots \end{bmatrix}. \tag{386}$$

If the probabilities β_n, $n \geqslant 0$ are arranged this way, one can indeed verify that the entries on the first row of $\mathbf{B}^n$ correspond to the mass function of the n-fold convolution of a_1, i.e. to the mass function of the number of 1-arrivals in n slots.

Secondly, once we know the distribution of q_0, we want to find out how many slots it will take before the virtual $\mathcal{P}$-queue collapses. Recall that the $\mathcal{P}$-queue will collapse at the end of a slot $J+k-1$ if it is empty then and if there are no 1-arrivals in that slot. Only in that situation we have that $q_k=0$. Let the random variable w indicate the number of slots after the A-period it takes for the virtual $\mathcal{P}$-queue to collapse, i.e. we define

$$w \triangleq \min\{k \geqslant 0 : q_k = 0\}\,. \tag{387}$$

In every slot, the virtual content will decrease by one unit as packets in the $\mathcal{P}$-queue move to the server and leave. However, the virtual content can also increase, due to 1-arrivals seizing a reservation. Specifically, as long as $q_k > 0$, we have that

$$q_{k+1} = q_k - 1 + a_{1,J+k}\,, \tag{388}$$

where the (conditional) distribution $\boldsymbol{\zeta}_i$ of q_0 is given by (385). Still assuming that $v=i$, the sequence q_k is a Markov Chain with starting probabilities $\boldsymbol{\zeta}_i$ and transition matrix $\mathbf{T}^*$ inferred from (388) as

$$\mathbf{T}^* = \left[\begin{array}{c|ccccc} 1 & 0 & 0 & 0 & \cdots \\ \hline \alpha & \beta_1 & \beta_2 & \beta_3 & \cdots \\ 0 & \alpha & \beta_1 & \beta_2 & \ddots \\ 0 & 0 & \alpha & \beta_1 & \ddots \\ 0 & 0 & 0 & \alpha & \ddots \\ \vdots & \vdots & \vdots & & \ddots \end{array}\right] = \left[\begin{array}{c|c} 1 & 0 \quad 0 \quad \cdots \\ \hline \mathbf{T}_0 & \mathbf{T} \end{array}\right]. \tag{389}$$

The variable w is then simply the number of transitions required to reach the state 0, if the distribution of the initial state is given by $\boldsymbol{\zeta}_i$. For further use, we also define the matrix $\mathbf{E}$ of appropriate size as

$$\mathbf{E} \triangleq \begin{bmatrix} 0 & 0 & 0 & 0 & \cdots \\ 1 & 0 & 0 & 0 & \cdots \\ 0 & 1 & 0 & 0 & \cdots \\ 0 & 0 & 1 & 0 & \cdots \\ \vdots & \vdots & & \ddots & \ddots \end{bmatrix} . \tag{390}$$

This matrix is an operator that shifts all elements in a row matrix one place to the left, such that e.g.

$$\boldsymbol{\zeta}_i \, \mathbf{E} \; = \; \begin{bmatrix} \zeta_{1,i} & \zeta_{2,i} & \zeta_{3,i} & \cdots & \end{bmatrix} \tag{391}$$

In fact, given that $v = i$, the distribution of w is commonly known as a phase-type (PH-type) distribution [152, p. 45] with representation $(\boldsymbol{\zeta}_i \mathbf{E}, \mathbf{T})$. In general, suppose the sequence q_k is a Markov chain with a number of possible states $q_k = 0, \ldots, M$ and transition matrix $\mathbf{T}^*$. Assume that one of the states, e.g. $q_k = 0$ is an *absorbing* state, which means that once the chain enters state 0, it will stay in this state forever. If such an absorbing state is present, then $\mathbf{T}^*$ has a block-structure as in (389), defining the sub-matrices $\mathbf{T}$ and $\mathbf{T}_0$. $\mathbf{T}$ contains the transition probabilities from and to states *other* than the absorbing state $q_k = 0$, while $\mathbf{T}_0$ is the column matrix with the transitions from any of the other states to absorption. A variable w is of PH-type if it represents the number of transitions before the chain reaches the absorbing state $q_k = 0$. This distribution not only depends on $\mathbf{T}$ and $\mathbf{T}_0$ but also on the initial state q_0. Applied to the case at hand, the distribution of the initial state is given by the row matrix $\boldsymbol{\zeta}_i$. As in [152], the mass function of w is now found to be

$$\begin{aligned} \text{Prob}[w = 0 | v = i] \; &= \; \zeta_{0,i} = \big(A_1(0)\big)^{i-1} = \alpha^{i-1} \, , \\ \text{Prob}[w = j | v = i] \; &= \; \boldsymbol{\zeta}_i \mathbf{E} \, \mathbf{T}^{j-1} \, \mathbf{T}_0 \, , \qquad\qquad j \geqslant 1 \, . \end{aligned} \tag{392}$$

For instance, if there are no 1-arrivals during the A-period then $q_0 = 0$ and the chain is in the absorbing state right from the beginning, which means that $w = 0$. On the other hand, if $q_0 > 0$ with initial probabilities $\boldsymbol{\zeta}_i \mathbf{E}$, it will remain in the set of non-absorbing states for $j - 1$ slots and finally be absorbed in the jth slot. The pgf of w follows from (392) as

$$\mathrm{E}[z^w | v = i] \; = \; \zeta_{0,i} + \boldsymbol{\zeta}_i (\mathbf{I} - z\mathbf{T})^{-1} \mathbf{T}_0 \, ,$$

still under the assumption that the A-period is $i - 1$ slots long.

The random variable w with the PH-type distribution derived above, represents the number of slots, starting with slot J, it takes for the virtual $\mathcal{P}$-queue to collapse. This means that the packet $\mathcal{P}$ will leave the system in slot $J + w$, and its total delay is $d_2 = v + w$. The quantity w is therefore what we have indicated as ‘additional’ delay, due to 1-arrivals seizing reservations in the (virtual) $\mathcal{P}$-queue *after* slot I.

All of the above is still under the assumption that the $\mathcal{P}$-queue has infinite size, which is obviously not the case in reality. Given that $v = i$, the *real* additional delay is not w but ν for which (378) and (379) hold. There are no more than N reservations in the $\mathcal{P}$-queue at the end of slot I that can be seized afterwards. The size of the $\mathcal{P}$-queue is exactly N at the end of the A-period and decreases by one every slot. Therefore, if the virtual $\mathcal{P}$-queue has not collapsed yet before slot $J+N$, the real $\mathcal{P}$-queue will do so anyway in that slot because $\mathcal{P}$ has shifted gradually to position 1. So whatever happens, $\mathcal{P}$ will be served no later than slot $J+N$ and we can conclude that

$$\nu \;=\; \min(N, w)\,. \tag{393}$$

Note that if more than N 1-arrivals occur during the A-period, they all will be accepted in the virtual $\mathcal{P}$-queue, but not in the real $\mathcal{P}$-queue. However, if this is the case then the upper Hessenberg structure of $\mathbf{T}$ ensures that $w > N$, such that (393) is not affected by this discrepancy.

The pgf $D_2(z)$ of the type 2 packet delay

With the conditional distribution of w and the relation (393), we now have sufficient information to construct the pgf $D_2(z)$ of the delay d_2 from (382). First, the conditional distribution of the additional delay ν follows from (392) and (393) as

$$\begin{aligned}
\text{Prob}[\nu\!=\!0|v\!=\!i] &= \zeta_{0,i} = \alpha^{i-1}\,,\\
\text{Prob}[\nu\!=\!j|v\!=\!i] &= \boldsymbol{\zeta}_i\mathbf{E}\mathbf{T}^{j-1}\mathbf{T}_0 = \mathbf{e}\mathbf{B}^{i-1}\mathbf{E}\mathbf{T}^{j-1}\mathbf{T}_0\,, \qquad 0<j<N\,,\\
\text{Prob}[\nu\!=\!N|v\!=\!i] &= \text{Prob}[w\!\geqslant\! N|v\!=\!i] = 1-\alpha^{i-1}-\mathbf{e}\mathbf{B}^{i-1}\mathbf{E}\sum_{j=0}^{N-2}\mathbf{T}^j\mathbf{T}_0\,,
\end{aligned} \tag{394}$$

which allows us to obtain the conditional pgf of the additional delay ν directly as

$$\begin{aligned}
\nu_i(z) &\triangleq \text{E}[z^{\nu}|v\!=\!i]\\
&= z^N + (1\!-\!z^N)\alpha^{i-1} + \mathbf{e}\mathbf{B}^{i-1}\mathbf{E}\Big(\sum_{j=1}^{N-1}(z^j - z^N)\mathbf{T}^{j-1}\Big)\mathbf{T}_0\,.
\end{aligned} \tag{395}$$

The *unconditional* distribution of the total delay $d_2\!=\!v\!+\!\nu$ can now be derived as

$$\begin{aligned}
D_2(z) \;&=\; \sum_{i=1}^{+\infty}\text{E}[z^{v+\nu}\{v\!=\!i\}] \;=\; \sum_{i=1}^{+\infty} z^i\,\nu_i(z)\,\text{Prob}[v\!=\!i]\\
&=\;\ldots\\
&=\; z^N V(z) + \frac{1\!-\!z^N}{\alpha}V(\alpha z) + z\,\mathbf{e}\,\hat{V}(z\mathbf{B})\,\mathbf{E}\Big(\sum_{j=1}^{N-1}(z^j - z^N)\mathbf{T}^{j-1}\Big)\mathbf{T}_0\,,
\end{aligned} \tag{396}$$

where we have used the pgf $V(z)$ of v, which is given by (268). The function $\hat{V}(z)$ is the pgf of $v\!-\!1$ and is defined in (381). Also, we denote by $\hat{V}(z\mathbf{B})$ a matrix function in z that is a power series of $z\mathbf{B}$ with the same coefficients as the power series $\hat{V}(z)$ in z. If we define $\hat{v}(n)$, $n\!\geqslant\!0$ to be the mass function of $v\!-\!1$, we have

$$\hat{V}(z) \;=\; \sum_{n=0}^{+\infty}\hat{v}(n)\,z^n\,, \qquad \text{and} \quad \hat{V}(z\mathbf{B}) \;=\; \sum_{n=0}^{+\infty}\hat{v}(n)\,(z\mathbf{B})^n\,. \tag{397}$$

In fact, what is needed in (396) is not the *whole* matrix $\hat{V}(z\mathbf{B})$, but only its first row $\mathbf{e}\,\hat{V}(z\mathbf{B})$. It is possible to give the generating function in x of the entries of $\mathbf{e}\,\hat{V}(z\mathbf{B})$ as follows. Recall from the previous section that the elements of the row matrix $\mathbf{e}\mathbf{B}^n$ have pgf $\big(A_1(x)\big)^n$. Therefore, if we expand $\hat{V}(z\mathbf{B})$ as in (397), then we know that the pgf of the row elements in the nth term is $\big(A_1(x)\big)^n$. The generating function in x of the entries in $\mathbf{e}\,\hat{V}(z\mathbf{B})$ therefore is

$$\hat{v}(0) + \hat{v}(1)A_1(x)z + \hat{v}(2)A_1^2(x)z^2 + \hat{v}(3)A_1^3(x)z^3 + \ldots = \hat{V}(zA_1(x)) \qquad (398)$$

In principle, we can use this result in expression (396) for the distribution of the type 2 delay. However, we will provide a better approach later on.

Convergence to AP: the case $N \to \infty$.

In the analysis of the type 1 delay in the previous section, we have observed that the delay distribution of the 1-packets converges for $N \to \infty$ to the delay as would be experienced in an absolute priority (AP) system. Obviously, we expect the same convergence towards AP-behaviour for the packets of type 2. In this section, we show that for an infinite number of reservations in the system, the pgf $D_2^{[\infty]}(z)$ is indeed the same as the pgf $D_2^{\mathrm{AP}}(z)$ obtained in [197].

The basic observation here is that in case $N \to \infty$, the $\mathcal{P}$-queue *effectively* contains an infinite number of reservations and therefore always has infinite size. Everything we have said for the *virtual* $\mathcal{P}$-queue is now also valid for the *real* $\mathcal{P}$-queue. Most importantly, the imposed restriction (393) on the additional delay ν becomes redundant now, i.e. we simply have that $\nu = w$ and consequently

$$d_2^{[\infty]} \;=\; v + w\,. \qquad (399)$$

The reason is that *all* 1-arrivals after slot I end up in the $\mathcal{P}$-queue in front of $\mathcal{P}$, as there are always available R's there. Before, only the first N 1-arrivals were admitted. The $\mathcal{P}$-queue will only collapse if there are no other packets left in front of $\mathcal{P}$ to choose for service. This collapse is caused by the fact that $\mathcal{P}$ jumps over the reservations into the server. Note that it is now impossible for $\mathcal{P}$ to shift gradually to position 1 because of the infinite barrier of R's.

Under the special condition of infinite reservations, the distribution of w can be derived in a much less complicated way than before. Let q_0 again be the content of the $\mathcal{P}$-queue just before slot J starts. The additional delay w can be represented as

$$w \;=\; \sum_{n=1}^{q_0} \eta_n\,, \qquad (400)$$

where η_n is the number of slots required for the content q_k to decrease from n to $n-1$. The time periods η_n are called *sub-busy periods* and in case all 1-arrivals are accepted to the $\mathcal{P}$-queue, their lengths are statistically independent. The pgf of η_n can be obtained by a recursive probabilistic argument due to [43, 196] that is illustrated in Fig. 4.18. Suppose that $q_k = n$ in a certain slot $J+k$ in which there are $a_{1,J+k}$ arrivals of type 1. Before the content can ever reach level $n-1$, the queue must first work away

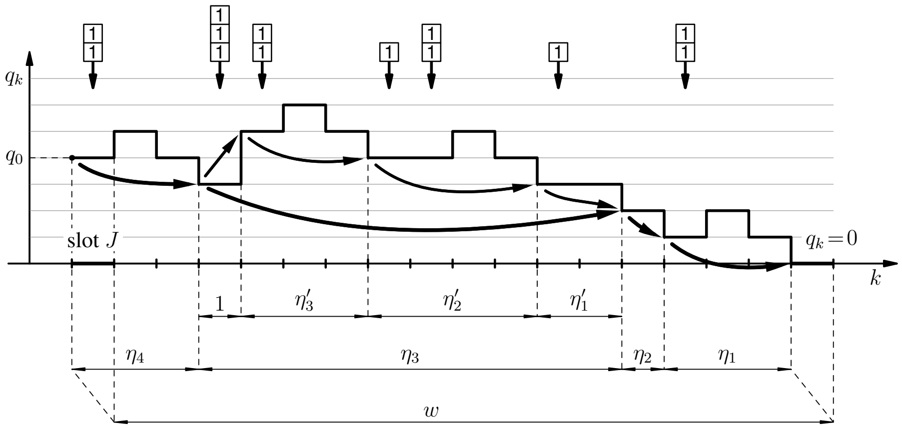


Figure 4.18: Evolution of the content q_k in the $\mathcal{P}$-queue. Just before slot J, the number of packets in the $\mathcal{P}$-queue is q_0. After slot J, the content must decrease by one q_0 times until it finally reaches zero and the $\mathcal{P}$-queue collapses. The additional delay w is composed of q_0 independent sub-busy periods η_n, $n=1,\ldots,q_0$.

these additional packets. Specifically, after slot $J+k$ the content must first decrease from $n+a_{1,J+k}-1$ to $n+a_{1,J+k}-2$, then to $n+a_{1,J+k}-3$ and so on until level $n-1$ is reached. Therefore, one can understand that each sub-busy period η_n *itself* consists of a number of (sub-)sub-busy periods $\eta'_{n'}$, i.e.

$$\eta_n \;=\; 1+\sum_{n'=1}^{a_{1,k}} \eta'_{n'}\,. \tag{401}$$

Both the periods η_n and $\eta'_{n'}$ represent the time required to decrease the content by one slot in a system with independent arrivals, so both periods are identical in the stochastic sense. Therefore, we can easily obtain the pgf $Y(z)$ of these sub-busy periods from (401) as

$$Y(z)=z\,A_1\big(Y(z)\big)\,, \qquad \text{and} \quad Y(1)=1\,. \tag{402}$$

The latter requirement is due to the fact that we want $Y(z)$ to be a proper pgf. Rouché's theorem on p. 159 can be used to show that there is always a unique function $Y(z)$ that satisfies (402).

As all 1-arrivals are admitted to the $\mathcal{P}$-queue, we know that its content q_0 just before the end of the A-period is equal to the number of 1-arrivals during this A-period. Given that $v=i$, the length of the A-period is $i-1$ slots, so we have

$$\mathrm{E}[z^{q_0}|v=i] \;=\; \big(A_1(z)\big)^{i-1}\,. \tag{403}$$

As the sub-busy periods η_n in the sum (400) are independent and identically distributed with common pgf (403), the distribution of w is

$$\mathrm{E}[z^{w}|v=i] \;=\; \big(A_1(Y(z))\big)^{i-1}\,. \tag{404}$$

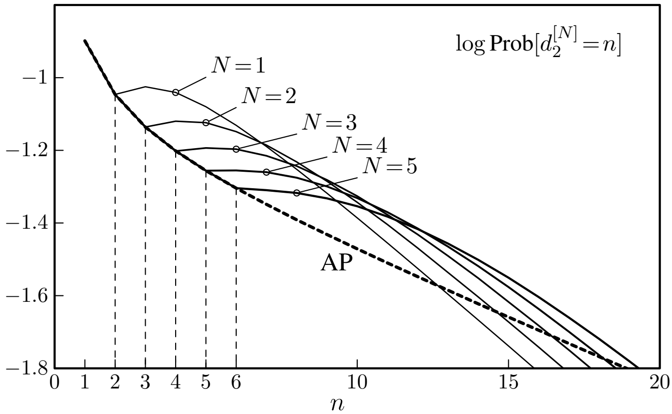


Figure 4.19: $\mathrm{Prob}[d_2^{[N]}=n]$ for $N=1,2,3,4,5$ compared to the corresponding type 2 delay distribution in case of AP.

The unconditional pgf $D_2^{[\infty]}$ of the delay $d_2^{[\infty]}$ in case there are an infinite number of R's in the system then follows as

$$\begin{aligned} D_2^{[\infty]}(z) &= \sum_{i=1}^{+\infty}\mathrm{E}[z^{i+w}\{v=i\}] = \sum_{i=1}^{+\infty} z^i\big(A_1(Y(z))\big)^{i-1}\mathrm{Prob}[v=i] \\ &= z\hat{V}(zA_1(Y(z))) = z\hat{V}(Y(z)) \\ &= \frac{p_0}{\lambda_2}\frac{A_T(Y(z))-A_1(Y(z))}{Y(z)-A_T(Y(z))} = D_2^{\mathrm{AP}}(z)\,. \end{aligned} \tag{405}$$

This is exactly the expression that was obtained for the delay of the type 2 packets in case of absolute priority, see [196]. Once again, we validated our assertion that the behaviour of the packets under the reservation discipline converges towards AP if N gets very large.

The function $Y(z)$ is closely related to the function $\hat{Y}(z)$ we have encountered in the analysis of the type 1 delay. Both functions are determined implicitly and by comparing (339) and (402), it is seen that

$$Y(z)=z\,\hat{Y}(z)\,. \tag{406}$$

Because of this relation, we now learn that $\hat{Y}(z)$ is a proper pgf as well. It is the distribution of $\eta-1$, where η is the length of a sub-busy period in a GI-1-1 system where the number of arrivals per slot are *iid* with distribution $A_1(z)$. In general, the presence of $Y(z)$ in expression (405) has a profound impact on the tail behaviour of the delay distribution. It turns out that the dominant singularity of $D_2^{\mathrm{AP}}(z)$ is not necessarily a pole but can also be a branch point. The dominant-pole approximation which we have used so often in this dissertation is therefore not applicable. The decay of $\mathrm{Prob}[d_2^{\mathrm{AP}}=n]$ for large n is no longer exponential but sub-exponential. For further discussion on the function $Y(z)$ and its impact on the tail distribution, we refer to [199] and the appendix of [123].

To conclude, we focus on a remarkable feature of the distribution of $d_2^{[N]}$ for *finite* N when compared to the corresponding delay under AP. In Fig. 4.19, we have plotted the mass function of the delay using the numerical inversion of our final expression for the pgf $D_2^{[N]}(z)$ to be developed later on. We show a logarithmic plot of $\text{Prob}[d_2^{[N]}=n]$ for systems with 1 up to 5 reservations and compare it to the mass function of the delay in case of AP. The considered arrival process is again that of (282) with $\lambda_1=\lambda_2=0.45$. From these curves, we make the following observation:

$$\text{Prob}[d_2^{[N]}=n] \;=\; \text{Prob}[d_2^{\text{AP}}=n]\,, \qquad n=1,2,\ldots N+1\,. \tag{407}$$

This can be understood as follows. First note that at the end of slot I, the prospects for $\mathcal{P}$ look exactly the same in the NR-system as under AP. Specifically, when $\mathcal{P}$ is stored in the queue, the number of packets to be served no later than $\mathcal{P}$ is v with pgf $V(z)$ under both disciplines. In the slots after I, the number of packets scheduled for service prior to $\mathcal{P}$ will *remain* identical in both systems as long as the 1-arrivals are stored in the $\mathcal{P}$-queue and not outside. In other words, the real and the virtual content of the $\mathcal{P}$-queue are the same as long as no 1-arrivals are stored *behind* $\mathcal{P}$. Then, observe also that it is *only* under these circumstances that $d_2^{[N]} \leqslant N+1$ can occur, which explains (407).

Matrices of finite dimension only

The main problem with expression (396) for the pgf of the type 2 delay in case there are N reservations in the system, is the fact that the matrices involved are of infinite dimension. This makes it hard to use this expression for practical computations. To make our point clear, let us define the row matrix $\mathbf{R}(z)$ and the column matrix $\mathbf{C}(z)$ as

$$\mathbf{R}(z) \;\triangleq\; \mathbf{e}\hat{V}(z\mathbf{B})\mathbf{E}\,, \tag{408}$$

$$\mathbf{C}(z) \;\triangleq\; \Big(\sum_{j=1}^{N-1}(z^j - z^N)\mathbf{T}^{j-1}\Big)\mathbf{T}_0\,, \tag{409}$$

so that the last term of (396) can be written simply as $z\mathbf{R}(z)\mathbf{C}(z)$. Since both factors are of infinite dimension, one is led to believe that the multiplication $\mathbf{R}(z)\mathbf{C}(z)$ requires the sum over an infinite number of terms. In reality however, this is a sum over only $N-1$ terms, because only the first $N-1$ entries of $\mathbf{C}(z)$ differ from zero. To see this, let us have a close look at the factors in (409). As only the first element of $\mathbf{T}_0$ is non-zero, see (389), the post-multiplication by $\mathbf{T}_0$ in fact selects the first column of the summation over j. Now, since the highest power of $\mathbf{T}$ occurring in this sum is $\mathbf{T}^{N-2}$, only the first $N-1$ entries on its first column are different from zero. This situation obviously arises due to the upper Hessenberg structure of $\mathbf{T}$. We conclude that in order to calculate $z\mathbf{R}(z)\mathbf{C}(z)$, we essentially need the first $N-1$ entries of $\mathbf{R}(z)$ and the first column of the powers $\mathbf{T}^j$ in $\mathbf{C}(z)$.

First, if we introduce the notation

$$\tilde{\beta}_i \triangleq \sum_{j=i}^{+\infty}\beta_j\,, \tag{410}$$

then we can define the $N{\times}N$ matrix $\tilde{\mathbf{T}}$ and the $N{\times}1$ column matrix $\tilde{\mathbf{T}}_0$ as the *truncated* versions of $\mathbf{T}$ and $\mathbf{T}_0$ respectively:

$$\tilde{\mathbf{T}} \triangleq \begin{bmatrix} \beta_1 & \beta_2 & \cdots & \beta_{N-1} & \tilde{\beta}_N \\ \alpha & \beta_1 & \cdots & \beta_{N-2} & \tilde{\beta}_{N-1} \\ 0 & \alpha & \cdots & \beta_{N-3} & \tilde{\beta}_{N-2} \\ \vdots & \ddots & \ddots & & \vdots \\ 0 & \cdots & 0 & \alpha & \tilde{\beta}_1 \end{bmatrix} \quad \text{and} \quad \tilde{\mathbf{T}}_0 \triangleq \begin{bmatrix} \alpha \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \tag{411}$$

Observe that $\tilde{\mathbf{T}}$ has the same upper Hessenberg structure as $\mathbf{T}$ and is still a stochastic matrix. Unlike $\mathbf{T}$ however, $\tilde{\mathbf{T}}$ is no longer Toeplitz. We can also compare the nth power of both $\mathbf{T}$ and $\tilde{\mathbf{T}}$ and observe that for the latter, we have the following structure,

$$\tilde{\mathbf{T}}^n = \begin{array}{cc} & \begin{array}{cccccc} & & & & \overbrace{}^{n} \end{array} \\ n{+}1\left\{\begin{array}{c} \\ \\ \\ \\ \end{array}\right. & \begin{bmatrix} * & * & \cdots & * & \star & \cdots & \star \\ * & * & \cdots & * & \star & \cdots & \star \\ \vdots & & & \vdots & \vdots & & \vdots \\ * & * & \cdots & * & \star & \cdots & \star \\ 0 & * & \cdots & * & \star & \cdots & \star \\ \vdots & \ddots & \ddots & \vdots & \vdots & & \vdots \\ 0 & \cdots & 0 & * & \star & \cdots & \star \end{bmatrix} \end{array}, \tag{412}$$

where $\star$ represents an entry that is affected by the truncation and therefore different than the corresponding entry in $\mathbf{T}^n$. The entries indicated by $*$ on the other hand are exactly the *same* as in $\mathbf{T}^n$. We clearly see that the first column of $\tilde{\mathbf{T}}^n$ remains unaffected as long as the power n does not exceed N. As the highest power of $\mathbf{T}$ that occurs in (409) is $\mathbf{T}^{N-2}$, we infer from (412) that only the first $N-1$ entries of

$$\Big(\sum_{j=1}^{N-1}(z^j - z^N)\tilde{\mathbf{T}}^{j-1}\Big)\tilde{\mathbf{T}}_0\,. \tag{413}$$

are nonzero and more importantly, are exactly the same as the corresponding entries in $C(z)$. Secondly, we can use a similar truncation for the matrix $\mathbf{B}$ in (408) because we are only interested in the first N elements of $\mathbf{e}\,\hat{V}(z\mathbf{B})$. In fact, the first of these elements is not used either due to the shifting operator $\mathbf{E}$. Let us define the $(N+1)\times(N+1)$ matrix $\tilde{\mathbf{B}}$ as

$$\tilde{\mathbf{B}} = \begin{bmatrix} \alpha & \beta_1 & \beta_2 & \cdots & \beta_{N-1} & \tilde{\beta}_N \\ 0 & \alpha & \beta_1 & \cdots & \beta_{N-2} & \tilde{\beta}_{N-1} \\ 0 & 0 & \alpha & \cdots & \beta_{N-3} & \tilde{\beta}_{N-2} \\ \vdots & & \ddots & \ddots & & \vdots \\ 0 & & & & \alpha & \tilde{\beta}_1 \\ 0 & 0 & 0 & \cdots & 0 & 1 \end{bmatrix}, \tag{414}$$

which is to be compared with (386). Because of the upper triangular structure of both $\mathbf{B}$ and $\tilde{\mathbf{B}}$, it is easily verified that the nth power of $\tilde{\mathbf{B}}$ results in

$$\tilde{\mathbf{B}}^n = \begin{bmatrix} * & * & * \cdots & * & \star \\ 0 & * & * \cdots & * & \star \\ 0 & 0 & * \cdots & * & \star \\ \vdots & & \ddots \; \ddots & \vdots & \vdots \\ 0 & & 0 & * & \star \\ 0 & & \cdots & 0 & 1 \end{bmatrix}, \tag{415}$$

for *any* $n \geqslant 1$. Here, the elements equal to the corresponding element in $\mathbf{B}^n$ are indicated by $*$ again whereas the elements that have changed due to the truncation are represented by $\star$. Specifically, if we look at the first row of $\tilde{\mathbf{B}}^n$, we see that the N first elements are the same as in $\mathbf{B}^n$. If we now also apply the operator $\mathbf{E}$ defined in (390) which is of order $(N+1)\times N$, we conclude that the first $N-1$ entries of $z\mathbf{R}(z)$ can equally be obtained as

$$z\,\mathbf{e}\hat{V}(z\tilde{\mathbf{B}})\,\mathbf{E}\,. \tag{416}$$

In expression (396) for the pgf $D_2(z)$, we can therefore replace the term $z\mathbf{R}(z)\mathbf{C}(z)$ by the product of (416) and (413). This yields

$$D_2(z) = z^N V(z) + \frac{1-z^N}{\alpha} V(\alpha z) + z\mathbf{e}\hat{V}(z\tilde{\mathbf{B}})\mathbf{E}\Big(\sum_{j=1}^{N-1}(z^j - z^N)\tilde{\mathbf{T}}^{j-1}\Big)\tilde{\mathbf{T}}_0\,. \tag{417}$$

In this expression, all matrices are of finite dimension, which makes it easier to derive practical results such as the expected value and the tail distribution of the delay. Nevertheless, one problem remains: if we evaluate $\hat{V}(z\tilde{\mathbf{B}})$ by using (397), we have to compute all positive powers of $\tilde{\mathbf{B}}$, which is of course not feasible. The following section provides a solution for this problem.

Spectral decomposition of $\tilde{\mathrm{B}}$

We now concentrate on the evaluation of the matrix function $\hat{V}(z\tilde{\mathbf{B}})$. From a purely mathematical point of view, one must be univocal about what is meant by the image of a matrix $\mathbf{A}$ under a *scalar* function $f(z)$. The result of such an operation is uniquely defined as long as the function $f(z)$ is defined on the spectrum $\sigma(\mathbf{A})$ or in other words, all eigenvalues of $\mathbf{A}$ must belong to the domain of $f(z)$. If such is the case, then it is possible to reduce the evaluation of $f(\mathbf{A})$ to the evaluation of f and its derivatives over the scalar set of eigenvalues $\sigma(\mathbf{A})$. This reduction is accomplished by what is generally known as the *spectral decomposition* of $\mathbf{A}$ and is described in [94, 141]. If the matrix $\mathbf{A}$ is diagonalisable, we already have discussed and applied the spectral decomposition technique in Sec. 3.A. However, in case of (414) we are dealing with a matrix that is in general *non-diagonalisable*, which requires a more general approach. Nevertheless, due to the special structure of this matrix we can still provide a decomposition that is generally valid. The use of spectral decomposition for evaluating the performance of queues dates back at least as far as 1974 [121]. Since then, it has successfully been applied in both continuous-time [21] and discrete-time [128, 129] models. In the latter, the entries of the decomposed matrix are themselves the z-transforms of stochastic variables. A rigorous and more mathematically inclined treatment of this subject is given in [92].

The first step in the decomposition of $\tilde{\mathbf{B}}$ is the identification of its eigenvalues. From (414), the characteristic polynomial of $\tilde{\mathbf{B}}$ is seen to be given by

$$\det(\tilde{\mathbf{B}} - \lambda \mathbf{I}) \;=\; (1-\lambda)(\alpha-\lambda)^N \tag{418}$$

Clearly, this means that $\tilde{\mathbf{B}}$ has only two eigenvalues 1 and α, with algebraic multiplicities 1 and N respectively. Both eigenvalues are in the analytic domain of $\hat{V}(z)$ under the condition (286), which establishes that the matrix $\hat{V}(z\tilde{\mathbf{B}})$ is uniquely defined. Note that $\tilde{\mathbf{B}}$ is a stochastic matrix, so it is no surprise to see that the largest eigenvalue is 1, as required by the Perron-Frobenius theorem. This eigenvalue is often referred to as the *PF-eigenvalue*. Also, if $\tilde{\mathbf{B}}$ is interpreted as a Markov transition matrix, the resulting chain has an absorbing state, namely the state with highest index $N+1$. The stationary distribution $\boldsymbol{\pi}$ of this chain is the solution of $\boldsymbol{\pi}\tilde{\mathbf{B}} = \boldsymbol{\pi}$ and is therefore a normalised left eigenvector of $\tilde{\mathbf{B}}$ corresponding to the PF-eigenvalue 1. As state $N+1$ is absorbing the chain is almost sure (with probability 1) to end up there after a while so we have

$$\boldsymbol{\pi} = \begin{bmatrix} 0 & 0 & \cdots & 0 & 1 \end{bmatrix}, \qquad \text{and} \quad \boldsymbol{\pi}' = \begin{bmatrix} 1 & 1 & \cdots & 1 & 1 \end{bmatrix}^T . \tag{419}$$

Here, $\boldsymbol{\pi}'$ represents a *right* eigenvector of $\tilde{\mathbf{B}}$ corresponding to eigenvalue 1, which is easily obtained from $\tilde{\mathbf{B}}\boldsymbol{\pi}' = \boldsymbol{\pi}'$. A matrix reduction that is applicable for all square matrices is the similarity transform $\mathbf{P}^{-1}\tilde{\mathbf{B}}\mathbf{P} = \mathbf{J}$ to the *Jordan normal form*. The Jordan form $\mathbf{J}$ of a matrix $\tilde{\mathbf{B}}$ is unique and every matrix similar to $\tilde{\mathbf{B}}$ has the same $\mathbf{J}$ (two square matrices are called *similar* if the one can be written as a similarity transform of the other and vice versa). In case of a matrix with the eigenstructure of $\tilde{\mathbf{B}}$, the Jordan reduction has the following block form

$$\mathbf{J} \;=\; \left[\begin{array}{c|ccc} 1 & & & \\ \hline & \mathbf{J}_\alpha^1 = \begin{bmatrix} \alpha & 1 & & & \\ & \alpha & 1 & & \\ & & \ddots & \ddots & \\ & & & \alpha & 1 \\ & & & & \alpha \end{bmatrix} & & \\ & & \mathbf{J}_\alpha^2 = \begin{bmatrix} \alpha & 1 & & \\ & \ddots & \ddots & \\ & & \alpha & 1 \\ & & & \alpha \end{bmatrix} & \\ & & & \ddots \\ & & & \mathbf{J}_\alpha^{g_\alpha} = \begin{bmatrix} \alpha & 1 & & \\ & \ddots & \ddots & \\ & & \alpha & 1 \\ & & & \alpha \end{bmatrix} \end{array}\right] . \tag{420}$$

As $\tilde{\mathbf{B}}$ has two distinct eigenvalues, J consists of two so-called *Jordan segments* $\mathbf{J}_1$ and $\mathbf{J}_\alpha$ with their size equal to the algebraic multiplicity of the corresponding eigenvalue. The 1×1 segment $\mathbf{J}_1$ simply has the eigenvalue 1 as its only element. In the $N\times N$ segment $\mathbf{J}_\alpha$ on the other hand, all elements on the diagonal are equal to the eigenvalue α and some of the entries directly *above* the diagonal are nonzero as well. These off-diagonal nonzero elements can always be chosen equal to 1 under similarity transform and their exact number and place is most revealing. Specifically, the Jordan segment $\mathbf{J}_\alpha$ can be split up into a number of *Jordan blocks* $\mathbf{J}_\alpha^1$, $\mathbf{J}_\alpha^2$, ... such that within each block there are only 1's directly above the diagonal. The number of Jordan blocks that can be discerned in the segment $\mathbf{J}_\alpha$ is called the *geometric multiplicity* g_α of the eigenvalue α and is equal to the dimension of the space spanned by the eigenvectors of α. Also, the size of the largest Jordan block $\mathbf{J}_\alpha^*$ is termed the *index* of eigenvalue α and is equal to the index of the matrix $\tilde{\mathbf{B}}-\alpha\mathbf{I}$. The index of a matrix is the smallest positive integer k for which the rank of $(\tilde{\mathbf{B}}-\alpha\mathbf{I})^k$ is equal to the rank of $(\tilde{\mathbf{B}}-\alpha\mathbf{I})^{k+1}$.

Note that if a square matrix is diagonalisable, all off-diagonal elements in its Jordan form are zero. Hence, each eigenvalue has index 1 and its geometric multiplicity equals the algebraic multiplicity. For a non-diagonalisable matrix $\mathbf{A}$ however, its Jordan normal form $\mathbf{J}$ is the closest we can get to diagonalisation by similarity transform. In general, if $\mathbf{A}$ has spectrum $\sigma(\mathbf{A})=\{\lambda_1,\lambda_2,\ldots,\lambda_s\}$, then for f defined on $\sigma(\mathbf{A})$, the matrix function $f(\mathbf{A})$ is defined as

$$f(\mathbf{A}) = \sum_{j=1}^{s}\sum_{i=0}^{k_j-1}\frac{1}{i!}f^{(i)}(\lambda_j)\,(\mathbf{A}-\lambda_j\mathbf{I})^i\,\mathbf{G}_j\,, \tag{421}$$

which is the extension of (206) to non-diagonalisable $\mathbf{A}$. In this expression, $f^{(i)}$ is the ith derivative of f and k_j denotes the index of eigenvalue λ_j. The matrix $\mathbf{G}_j$ is the *spectral projector* or *constituent* belonging to eigenvalue λ_j for which the properties on p. 120 still apply. In general, these projectors can be calculated from the similarity transform matrix $\mathbf{P}$ for which $\mathbf{P}^{-1}\mathbf{A}\mathbf{P}=\mathbf{J}$. If $\mathbf{P}$ is partitioned conformably as

$$\mathbf{A} = \mathbf{P}\mathbf{J}\mathbf{P}^{-1} = \begin{bmatrix}\mathbf{P}_1 & \mathbf{P}_2 & \cdots & \mathbf{P}_s\end{bmatrix}\begin{bmatrix}\mathbf{J}_1 & & & \\ & \mathbf{J}_2 & & \\ & & \ddots & \\ & & & \mathbf{J}_s\end{bmatrix}\begin{bmatrix}\mathbf{Q}_1\\ \mathbf{Q}_2\\ \vdots\\ \mathbf{Q}_s\end{bmatrix}. \tag{422}$$

with $\mathbf{J}_j$ the Jordan segment corresponding to eigenvalue λ_j, then $\mathbf{G}_j=\mathbf{P}_j\mathbf{Q}_j$. We also note that the columns of $\mathbf{P}_j$ span the space of the right eigenvectors corresponding to λ_j while the rows of $\mathbf{Q}_j$ span the space of the left eigenvectors. An additional property worth mentioning is

$$(\mathbf{A}-\lambda_j\mathbf{I})^k\mathbf{G}_j = \mathbf{0} \qquad \text{for } k \geqslant k_j = \text{index}(\lambda_j) \text{ and } j = 1,\ldots,s\,. \tag{423}$$

In the case of the matrix $\tilde{\mathbf{B}}$ in (414), $\lambda = 1$ is a *simple* eigenvalue, meaning that both its algebraic and geometric multiplicity are 1. For such eigenvalues, the projector can be calculated easily from an arbitrary left and right eigenvector (e.g. $\boldsymbol{\pi}$ and $\boldsymbol{\pi}'$) as follows:

$$\mathbf{G}_1 = \frac{\boldsymbol{\pi}'\boldsymbol{\pi}}{\boldsymbol{\pi}\boldsymbol{\pi}'} = \begin{bmatrix} 0 & 0 & \cdots & 0 & 1 \\ 0 & 0 & \cdots & 0 & 1 \\ \vdots & \vdots & & & \vdots \\ 0 & 0 & \cdots & 0 & 1 \end{bmatrix} . \tag{424}$$

To obtain the second projector $\mathbf{G}_\alpha$ corresponding to eigenvalue α, we normally require the transform matrix $\mathbf{P}$ and its partitioning as in (422). However, we can also use the fact that all projectors must sum up to $\mathbf{I}$, such that

$$\mathbf{G}_\alpha = \mathbf{I} - \mathbf{G}_1 = \begin{bmatrix} 1 & 0 & \cdots & 0 & -1 \\ 0 & 1 & \cdots & 0 & -1 \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & & 1 & -1 \\ 0 & 0 & \cdots & 0 & 0 \end{bmatrix} . \tag{425}$$

Under the most general assumptions for the arrival distribution $A_1(z)$ of the 1-packets, and hence for the entries of $\tilde{\mathbf{B}}$, the Jordan segment $\mathbf{J}_\alpha$ has only one block. The size of this block is obviously N which is therefore also the index k_α of α. For the function $f(\tilde{\mathbf{B}}) = \hat{V}(z\tilde{\mathbf{B}})$, (421) then becomes

$$\hat{V}(z\tilde{\mathbf{B}}) = \hat{V}(z)\mathbf{G}_1 + \sum_{i=0}^{N-1} \frac{1}{i!} f^{(i)}(z\alpha)\,(\tilde{\mathbf{B}} - \alpha\mathbf{I})^i\,\mathbf{G}_\alpha\,. \tag{426}$$

Nevertheless, should for some specific $A_1(z)$ the index of α be lower than N, then (426) is still valid due to (423). If we apply this result into expression (417) for the pgf of the type 2 delay, we get

$$\begin{aligned} D_2(z) &= z^N V(z) + \frac{1-z^N}{\alpha} V(\alpha z) \\ &\quad + z\,\hat{V}(z)\mathbf{e}\,\mathbf{G}_1\mathbf{E}\Big(\sum_{j=1}^{N-1}(z^j - z^N)\tilde{\mathbf{T}}^{j-1}\Big)\tilde{\mathbf{T}}_0 \\ &\quad + z\sum_{i=0}^{N-1}\frac{z^i}{i!}\hat{V}^{(i)}(\alpha z)\mathbf{e}\,(\tilde{\mathbf{B}} - \alpha\mathbf{I})^i\mathbf{G}_\alpha\mathbf{E}\Big(\sum_{j=1}^{N-1}(z^j - z^N)\tilde{\mathbf{T}}^{j-1}\Big)\tilde{\mathbf{T}}_0\,. \end{aligned} \tag{427}$$

The second line in this expression is always equal to zero. To see why, observe from (424) that

$$\mathbf{e}\,\mathbf{G}_1\mathbf{E} = \begin{bmatrix} 0 & \cdots & 0 & 1 \end{bmatrix}, \quad \text{and} \quad \tilde{\mathbf{T}}_0 = \begin{bmatrix} \alpha & 0 & \cdots & 0 \end{bmatrix}^T . \tag{428}$$

Therefore, pre-multiplying by $\mathbf{e}\,\mathbf{G}_1\mathbf{E}$ and post-multiplying by $\tilde{\mathbf{T}}_0$ in fact only selects the first entry on the Nth row of the summation in between. As the term with the highest power in the sum is $\tilde{\mathbf{T}}^{N-2}$, we see from (412) that this entry is zero. Our final result for the pgf of the delay of type 2 now is

$$D_2(z) = z^N V(z) + \frac{1-z^N}{\alpha} V(\alpha z) \tag{429}$$

$$+ z \sum_{i=0}^{N-1} \frac{z^i}{i!} \hat{V}^{(i)}(\alpha z) \mathbf{e}\, (\tilde{\mathbf{B}} - \alpha \mathbf{I})^i \mathbf{G}_\alpha \mathbf{E} \Big(\sum_{j=1}^{N-1} (z^j - z^N) \tilde{\mathbf{T}}^{j-1} \Big) \tilde{\mathbf{T}}_0 \,.$$

In the following sections, we use this expression to calculate the mean value and the tail distribution of $d_2^{[N]}$, the type 2 packet delay in a system with N reservations.

Mean value of the type 2 packet delay

In principle, the moments of d_2 up to any order can be obtained from its pgf $D_2(z)$ by invoking the moment-generating property. For instance, the mean value $\mathrm{E}[d_2]$ follows as the first derivative of $D_2(z)$ evaluated for $z = 1$. From (429), we thus find

$$\begin{aligned} \mathrm{E}[d_2] \;&=\; N + V'(1) - N\hat{V}(\alpha) \\ &+ \sum_{i=0}^{N-1} \frac{1}{i!} \hat{V}^{(i)}(\alpha)\, \mathbf{e}\, (\tilde{\mathbf{B}} - \alpha \mathbf{I})^i \, \mathbf{G}_\alpha \mathbf{E} \Big(\sum_{j=1}^{N-1} (j - N) \tilde{\mathbf{T}}^{j-1} \Big) \tilde{\mathbf{T}}_0 \,, \end{aligned} \tag{430}$$

where $V'(1)$ and $\hat{V}(\alpha)$ easily follow from (268) and (381) as

$$V'(1) \;=\; 1 + \frac{(1-\lambda_1)\lambda_T' - (1-\lambda_T)\lambda_1'}{2(1-\lambda_T)\lambda_2} \,, \tag{431}$$

$$\hat{V}(\alpha) \;=\; \frac{1-\lambda_T}{\lambda_2} \, \frac{A_T(\alpha) - A_1(\alpha)}{\alpha - A_T(\alpha)} \,. \tag{432}$$

If N is large, probably the most time-consuming part of calculating the mean delay as in (430), is obtaining the derivatives of $\hat{V}(z)$. If the joint pgf $A(z_1, z_2)$ is complicated, so will the pgf $\hat{V}(z)$ and it may be difficult to provide the subsequent derivatives $\hat{V}^{(i)}(z)$ in a symbolical way for large i within reasonable time. In such cases, it is perhaps a good idea to revert to the probability domain. Specifically, from the power expansion of $\hat{V}(z)$ in (397), it is seen that

$$\frac{1}{i!} \hat{V}^{(i)}(\alpha) \;=\; \sum_{k=i}^{+\infty} C_k^i \, \alpha^{k-i} \hat{v}(k) \,, \qquad i \geqslant 0 \,, \tag{433}$$

where $C_k^i = k!/i!(k-i)!$ indicates the number of subsets with i elements in a set of k elements. The quantities $\hat{V}^{(i)}(\alpha)$ can then be approximated by adding terms in (433) until a certain precision is attained.

As was the case with the delay d_1 of the 1-packets, it is now also possible to calculate the mean delay of the type 2 packets in a progressive way with respect to the number of reservations. The following algorithm economically arranges the required matrix multiplications to provide the mean values $\mathrm{E}[d_2^{[n]}]$ for systems with 1 up to N reserved spaces simultaneously. It is based on the following observation. Instead of truncating $\mathbf{B}$ to the square matrix $\tilde{\mathbf{B}}$ of size $N+1$, we might as well have truncated to a matrix of any order *higher* than $N+1$. In the same way, truncation of $\mathbf{T}$ to $\tilde{\mathbf{T}}$ of any size larger than N would yield exactly the same result for $D_2^{[N]}(z)$ and hence also for $\mathrm{E}[d_2^{[N]}]$. Therefore, the mean values can be computed as

- Calculate $V'(1)$ and $\hat{V}(\alpha)$ from (431) and (432).
- If the approximation (433) is used, obtain the coefficients $\hat{v}(k)$ in the power expansion (397) of $\hat{V}$ by numerically inverting the pgf $\hat{V}(z)$.
- Initialise the matrices

$$\mathbf{Q}_0 = \mathbf{0}\,, \qquad \mathbf{F}_0 = \mathbf{0}\,, \qquad \mathbf{C}_0 = \mathbf{I}\,, \tag{434}$$

  which have dimensions $N\times N$, $1\times(N+1)$ and $(N+1)\times(N+1)$ respectively.
- For $n=1,\ldots,N$ subsequently update these matrices as

$$\begin{aligned}
\mathbf{Q}_n &= \mathbf{Q}_{n-1}\,\tilde{\mathbf{T}} - n\,\mathbf{I}\,,\\
\mathbf{F}_n &= \mathbf{F}_{n-1} + \frac{1}{(n-1)!}\hat{V}^{(n-1)}(\alpha)\,\mathbf{e}\,\mathbf{C}_{n-1}\,,\\
\mathbf{C}_n &= \mathbf{C}_{n-1}\,(\tilde{\mathbf{B}} - \alpha\mathbf{I})\,.
\end{aligned}$$

  For each n, the mean delay of a type 2 packet then follows as

$$\mathrm{E}[d_2^{[N]}] \;=\; n + V'(1) - n\,\hat{V}(\alpha) + \mathbf{F}_n\mathbf{G}_\alpha\mathbf{E}\mathbf{Q}_{n-1}\tilde{\mathbf{T}}_0\,.$$

Tail distribution of the type 2 packet delay

To obtain the tail probabilities of the type 2 delay, we can use the dominant-pole approximation again, i.e.

$$\mathrm{Prob}[d_2^{[N]}=n] \;\cong\; -\theta_2^{[N]}\,z_d^{-n-1}\,, \tag{435}$$

which is very accurate if n is sufficiently large. Consequently, we have to identify the dominant pole z_d of $D_2^{[N]}(z)$ and then obtain its residue $\theta_2^{[N]}$ in the point $z=z_d$. It is not difficult to see that the smallest real positive pole of expression (429) can only be due to the first term, i.e. it is the smallest root outside the unit disc in the denominator of $V(z)$. Therefore, the dominant pole z_d can be calculated numerically from (364) again and is the *same* as the dominant pole of the type 1 delay, as expected. The contributions in the other terms of $D_2^{[N]}(z)$ are either purely polynomial, or they have the form $\hat{V}^{(i)}(\alpha z)$, $i\geqslant 0$. While the polynomial parts have no singularities at all, the functions $\hat{V}^{(i)}(\alpha z)$ have the same singularities as $V(\alpha z)$, which are obviously larger than those of $V(z)$ since $\alpha<1$. For the residue $\theta_2^{[N]}$ we find

$$\begin{aligned}
\theta_2^{[N]} &= \lim_{z\to z_d}(z-z_d)D_2(z) \;=\; \lim_{z\to z_d}(z-z_d)z^N V(z)\\
&= z_d^N\,\frac{p_0}{\lambda_2}\,z_d(z_d - A_1(z_d))\,\lim_{z\to z_d}\frac{z-z_d}{z-A_T(z)}\\
&= \frac{p_0}{\lambda_2}\,\frac{z_d-1}{1-A_T'(z_d)}\,z_d^{N+1}\,,
\end{aligned} \tag{436}$$

where we have used expression (268) for the pgf $V(z)$. This result is surprisingly simple and similar to the result we found in the analysis of the system with only one reserved space. The residues $\theta_2^{[n]}$ for systems with $n=2,3,\ldots$ reservations follow from the residue $\theta_2^{[1]}$ given in (281) simply by multiplying with z_d^{n-1}. On a logarithmic plot, this means that the tail distributions (435) of the delay $d_2^{[n]}$, $n=1,2,3,\ldots$ will be equidistant straight lines with a vertical distance of $\log z_d$ between them, as will be demonstrated in the examples further on.

Bounds for the delay of type 2

In fact, it is no surprise that the asymptotic behaviour of the probabilities $\text{Prob}[d_2=n]$ is determined by the term $z^N V(z)$. Recall the bounds (379) for d_2, i.e.

$$v \leqslant d_N^{[2]} \leqslant v+N\,, \tag{437}$$

where the delay is equal to the lower bound if *none* of the reservations is seized during the A-period. On the other hand, the delay is equal to the upper bound in case *all* N reservations are seized during the waiting time of $\mathcal{P}$. Taking the generating functions of both bounds we have

$$D_{2,\text{low}}^{[N]}(z) \;=\; V(z)\,, \qquad \text{and} \qquad D_{2,\text{up}}^{[N]}(z) \;=\; z^N V(z)\,. \tag{438}$$

The higher the value of v, the longer also the A-period and the number of 1-arrivals during the A-period that seize reservations. So if v is high, the more likely the event that all N reservations are seized and that the delay attains its upper bound $v+N$. Therefore, it is safe to say that packets with very high delay almost always experience the maximum delay given by the upper bound. One can also use the bounds (438) to provide a rough estimate of the mean delay of the 2-packets. Applying the moment-generating property for (438), we find

$$V'(1) \;\leqslant\; \mathrm{E}[d_2^{[N]}] \;\leqslant\; V'(1)+N\,. \tag{439}$$

where $V'(1)$ is given in (431). These bounds can be useful especially if N is not too large, as will be demonstrated in the next Section. For very large N, instead of the upper bound in (439) one can use the mean delay in case of AP as an upper bound, which is easier to calculate than the exact value. For intermediate N however, the values must be calculated exactly.

4.3.6 Discussion of results: some examples

In the discussion above, we have presented specific algorithms to calculate the mean value and tail distribution of both d_1 and d_2, the delay experienced by an arbitrary packet of type 1 or type 2 respectively. In this section, we use these algorithms to provide a number of worked-out numerical examples so as to illustrate some specific aspects of the packet delay distribution under the Reservation discipline. Where appropriate, the results are compared to the corresponding results under FIFO and AP.

Let us first choose the distribution of the arrivals as in (282), i.e. the number of arrivals of type 1 and type 2 within a slot have a geometric and Poisson distribution respectively and are independent. The mean number of arrivals is λ_1 and λ_2 for type 1 and type 2 respectively and the system experiences a total load of $\lambda_T=\lambda_1+\lambda_2$.

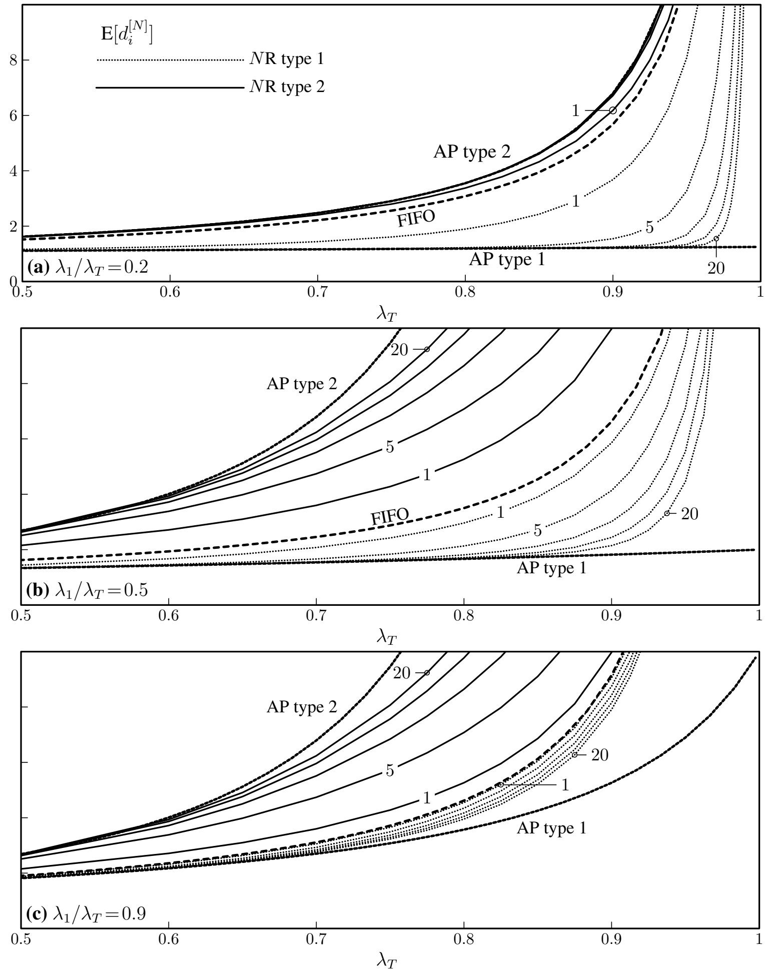


Figure 4.20: Mean delay of both 1- and 2-packets as a function of the total load λ_T for various values of $N = 1, 5, 10, 15, 20$ and in case the arrivals are distributed as in (282). The traffic mix is $\lambda_1/\lambda_T = 0.2$ in **(a)**, 0.5 in **(b)** and 0.9 in **(c)**. The corresponding curves for FIFO and AP are shown as well.

In Fig. 4.20, the mean value of $d_1^{[N]}$ and $d_2^{[N]}$ is plotted as a function of the load λ_T. In each of the plots **(a)**, **(b)** and **(c)**, the mean delays of both types are shown for $N = 1, 5, 10, 15$ and 20 reservations in the queue. The mean delay under FIFO and the mean delay for 1- and 2-packets under AP are shown as well. As the FIFO-discipline makes no distinction between the two types of packets we have plotted only *one* curve, indicating the delay of an arbitrary packet regardless of its type. The first thing we notice in these and all of the following plots, is that the curves for $\mathrm{E}[d_1^{[N]}]$ are always positioned between those for $\mathrm{E}[d_1^{\mathrm{AP}}]$ and the FIFO value $\mathrm{E}[d^{\mathrm{FIFO}}]$. Similarly, the curves

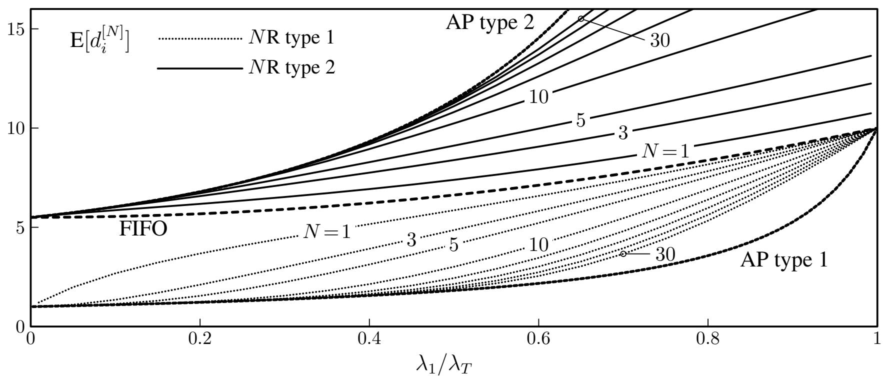


Figure 4.21: Mean delay of both 1- and 2-packets versus the traffic mix λ_1/λ_T in case of $N = 1, 3, 5, 10, 15, 20, 25$ and 30 reservations in the queue. The arrivals of type 1 and type 2 are independent and have a geometric and Poisson distribution respectively with total load $\lambda_T = 0.9$.

for $E[d_2^{[N]}]$ are positioned between $E[d^{\text{FIFO}}]$ and $E[d_2^{\text{AP}}]$, i.e.

$$E[d_1^{\text{AP}}] \leqslant E[d_1^{[N]}] \leqslant E[d^{\text{FIFO}}] \leqslant E[d_2^{[N]}] \leqslant E[d_2^{\text{AP}}]\,, \tag{440}$$

where strict equality only holds for some very specific arrival process. As N increases, the mean delay under the Reservation discipline tends more and more towards the extreme values under AP. Secondly, we observe that the relative position of the curves depends highly on the fraction of 1-packets present in the arriving traffic, i.e. on λ_1/λ_T. In **(a)**, this traffic mix is equal to 0.2 and the curves for the type 2 delay are very close together: only $E[d_2^{[1]}]$ can be distinguished from the AP-curve while for higher N, the curves almost coincide in the shown region. This indicates that if the traffic consists for the most part of 2-packets, the type 2 delay will deviate only slightly from the FIFO-value, which is intuitively clear. Conversely, if the traffic consists for the largest part of 1-packets, the curves for the type 1 delay lie very close together as is shown in **(c)**, where the traffic mix is 0.9. Indeed, as there are only few 2-packets in this case, the delay reduction that a 1-packet can realise by jumping over 2-packets in the queue is rather limited. In **(b)** the traffic mix is 0.5 resulting in a more balanced differentiation of the delay between the two types. Finally, in all three plots we see that as the total load increases, it takes a larger number of reservations to reach the AP-limit. However, no matter how many reservations are present in the system, the mean delay will *always* move away from the AP-curve if only the load λ_T gets high enough.

These observations are visually more apparent in Fig. 4.21 where we have plotted $E[d_1^{[N]}]$ and $E[d_2^{[N]}]$ as functions of the traffic mix λ_1/λ_T and for a fixed total load $\lambda_T = 0.9$. Again, the higher N, the more the mean delay of both types deviates from FIFO and the closer they get to their respective AP limits. On the far left side of the plot, there are but few 1-packets among a multitude of 2-packets. As a consequence,

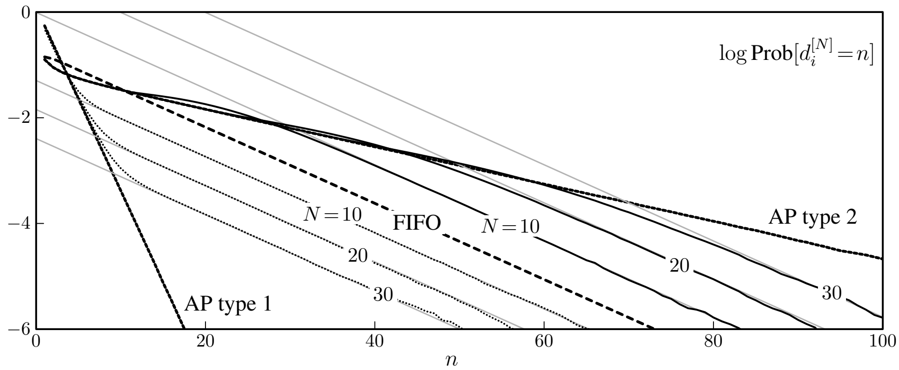


Figure 4.22: $\log \mathrm{Prob}[d_i^{[N]} = n]$ of the delay for both types of packets in case there are $N = 10, 20$ and 30 reservations. The load of the system is 0.9, equally shared among both traffic streams. The corresponding delay distributions for AP and FIFO are also shown.

the queue contains mainly 2-packets and always has almost all of its reservations positioned directly in front of the server. Therefore, a rare arriving 1-packet can generally jump over the whole queue content and be served directly in the next slot. So even if there is only one reserved space, the behaviour under the Reservation discipline is equal to that under AP for very low λ_1/λ_T, resulting in a maximal delay differentiation. On the far right of the plot, most of the traffic is of type 1, while 2-packets arrive only very rarely. From the point of view of the 1-packets, there is no difference between FIFO, AP or the intermediate Reservation discipline if λ_1/λ_T is very high. However, the delay of a rare 2-packet is influenced a great deal by the service discipline in this case. While such a packet is almost sure to stay in the queue forever under AP, its delay under the Reservation disciplines increases from the FIFO value more or less linearly with N. In our opinion, this is where the main strength of the Reservation discipline compared to AP emerges. If only a small part of the traffic consists of low-priority traffic, their delay can be chosen at an arbitrary level by changing N, whereas the delay is infinite under AP (packet starvation).

In Fig. 4.22 a logarithmic plot is shown of the delay mass function for $\lambda_1 = \lambda_2 = 0.45$ and for $N = 10, 20, 30$. Here, the curves for $\mathrm{Prob}[d_1^{[N]} = n]$ and $\mathrm{Prob}[d_2^{[N]} = n]$ under the Reservation discipline were obtained by simulation, as well as the reference curves for FIFO and AP. The gray lines indicate the result of the dominant-pole approximation of the tail distribution and are seen to fit very accurately for high n. The same observation as in the previous plots can be made here: the mass functions tend more and more towards the AP-limits as N increases. In fact, the following qualitative observation holds for the delay of both types $i = 1, 2$. For low values of n, the mass function of $d_i^{[N]}$ decays at the same rate as the corresponding probabilities for d_i^{AP}. If on the other hand n is high, the mass function will decay at the same rate as the tail distribution of d^{FIFO}. For intermediate n, the curve for $\mathrm{Prob}[d_i^{[N]} = n]$ is in transition between the AP and FIFO behaviour. Note that for $n \leqslant N + 1$, the correspondence

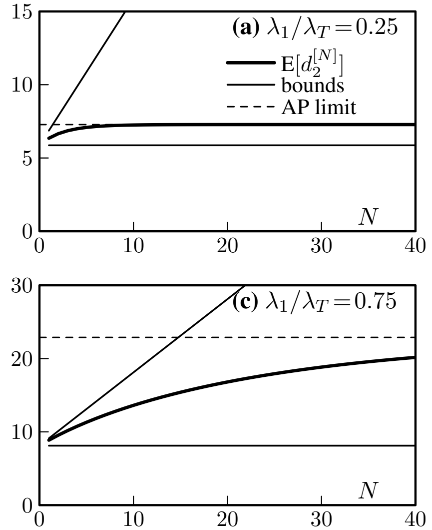


Figure 4.23: Plots of the mean type 2 delay $\mathrm{E}[d_2^{[N]}]$ versus N in case of the arrival process (282), for load $\lambda_T = 0.9$. The traffic mix is 0.25 in **(a)**, 0.5 in **(b)**, 0.75 in **(c)** and 0.9 in **(d)**. The upper and lower bounds as well as the AP-limit are also indicated.

between $\mathrm{Prob}[d_2^{\mathrm{AP}} = n]$ and $\mathrm{Prob}[d_2^{[N]} = n]$ is exact, as we have explained in (407).

In Fig. 4.23 and Fig. 4.24 we concentrate on the bounds (438) and (439) that we have established in the analysis of the type 2 delay. We try to find out in what circumstances they can be useful by comparing these bounds to the exact values of the mean delay $\mathrm{E}[d_2^{[N]}]$ and mass function $\mathrm{Prob}[d_2^{[N]} = n]$. In Fig. 4.23 we have chosen $\lambda_T = 0.9$ and plotted the mean delay as a function of the number of reservations N, for increasing values of the traffic mix λ_1/λ_T. In each of the plots, we included the lower bound $V'(1)$ and upper bound $V'(1)+N$, as well as the AP limiting value which is reached for $N \to \infty$. Obviously, the bounds are tighter if N is small but become useless for high N as the upper bound increases linearly and eventually even exceeds the AP-limit. In **(a)**, the traffic mix is 0.25 and we see that the bounds are very far apart. However, as the traffic mix increases to 0.5 in **(b)**, to 0.75 in **(c)** and to 0.9 in **(d)**, we see that for low N, the upper bound comes closer to the actual value of $\mathrm{E}[d_2^{[N]}]$. The value of the load λ_T is important as well: only if the load is high enough will $V'(1)$ be of higher magnitude than N, making the upper and lower bound relatively closer. We can conclude that $\mathrm{E}[d_2^{[N]}]$ can be roughly approximated by the upper bound $V'(1)+N$ only if N is rather small and both the total load and the traffic mix are high.

In Fig. 4.24 we show a logarithmic plot of $\mathrm{Prob}[d_2^{[N]} = n]$ for a fixed load $\lambda_T = 0.9$ and for the traffic mix increasing from 0.25 in **(a)** to 0.9 in **(d)**. In each case, we considered both $N = 5$ and $N = 10$. In these plots, the mass function of $d_2^{[N]}$ was obtained by numerical inversion of (429) using the technique explained in [20]. In each of the plots, we have also shown the numerically inverted bounds in (438). As was explained on p. 211, the tail of the type 2 delay always tends towards the

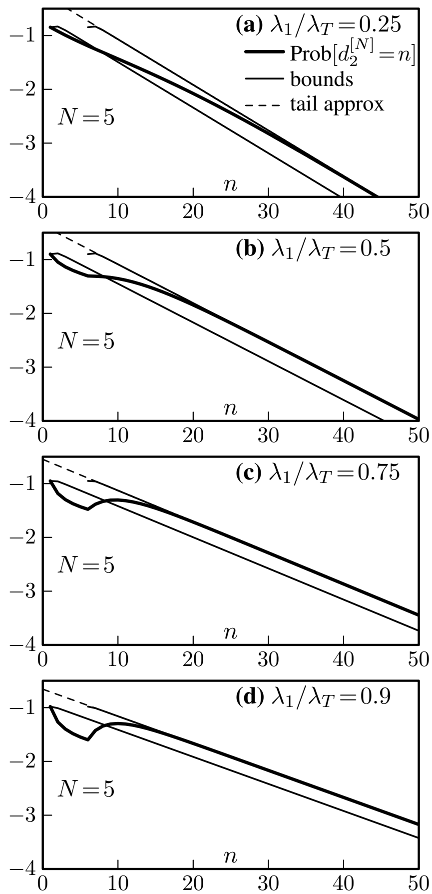


Figure 4.24: Logarithmic plots of Prob$[d_2^{[N]} = n]$ obtained by numerically inverting $D_2^{[N]}(z)$ in case of arrival process (282) and for $\lambda_T = 0.9$. The traffic mix changes from 0.25 in **(a)** to 0.9 in **(b)** and in each case both $N = 5$ and $N = 10$ is considered. For $n > N+1$ the curve changes shape and slowly converges to the stochastic upper bound given by the mass function of $z^N V(z)$.

upper bound. Note that the bounds $V(z)$ and $z^N V(z)$ which are plotted here in the probability domain are *stochastic* bounds and do not require that the mass function of $d_2^{[N]}$ lies between the two curves in all points. The specific shape of the mass function Prob$[d_2^{[N]} = n]$ for intermediate $n > N+1$ (when the transition from AP to FIFO decay takes place) is influenced by the value of λ_1/λ_T as demonstrated in Fig. 4.24.

We have also considered an example for which the arrivals of both types within a slot are *not* independent. Inspired on [196, 197], let the joint pgf of the numbers of 1- and 2-arrivals per slot be

$$A(z_1, z_2) \;=\; \left[1 - \frac{\lambda_1}{M}(1-z_1) - \frac{\lambda_2}{M}(1-z_2)\right]^M . \tag{441}$$

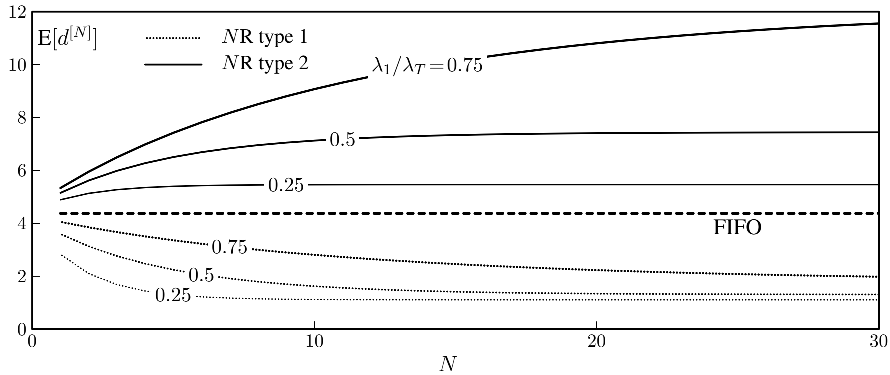


Figure 4.25: Mean value of the delay experienced by packets of type 1 and type 2 as a function of N in case of arrival process (441), a total load of $\lambda_T = 0.9$ and traffic mix $\lambda_1/\lambda_T = 0.25, 0.5, 0.75$. The mean delay in case of FIFO is also indicated for reference.

Note that the number of 1-arrivals per slot on the one hand and the number of 2-arrivals on the other are both binomially distributed but are (negatively) correlated because no more than M packets can arrive in one slot. In Fig. 4.25 we show the mean delays $\mathrm{E}[d_1^{[N]}]$ and $\mathrm{E}[d_2^{[N]}]$ of both types versus the number of reservations N in case the total load is 0.9 and for three values of the traffic mix: $\lambda_1/\lambda_T = 0.25, 0.5, 0.75$. The middle curve shows the mean FIFO delay of an arbitrary packet. Again, we see that the mean delay of both types saturates (to the AP-limit) as N increases. As before, we see that this saturation sets in faster if the fraction of 1-packets in the traffic is lower. The influence of the traffic mix is shown more clearly in Fig. 4.26 which is the counterpart of Fig. 4.21 for arrival distribution (441). The same general conclusions can be drawn here. Note that in contrast to Fig. 4.21 however, the FIFO curve is completely flat. This is due to the fact that the total number of arrivals per slot always has distribution

$$A_T(z) = A(z,z) = \left[1 - \frac{\lambda_T}{M}(1-z)\right]^M , \tag{442}$$

regardless of how the load is distributed between the two types.

4.4 Conclusion

We have considered a discrete-time queue operating under the Reservation discipline. The packets that arrive to the queue belong to one of two types reflecting their QoS (Quality of Service) requirements: packets of type 1 are more delay-sensitive than the packets of type 2. The Reservation discipline realises a delay differentiation between the two types by reserving positions in the queue for future arrivals of type 1 and consists of the following set of simple rules:

- Within a slot, the 1-packets are always stored in the queue *prior* to the 2-packets.

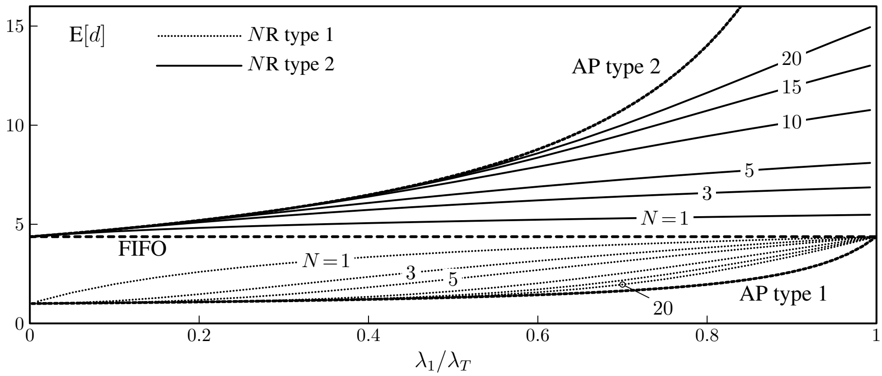


Figure 4.26: Mean delay of both 1- and 2-packets versus the traffic mix λ_1/λ_T in case of $N = 1, 3, 5, 10, 15$ and 20 reservations in the queue. The arrivals have a joint binomial distribution as in (441).

- ► When stored, a 1-packet seizes the position of the most advanced reservation in the queue and creates a *new* reservation on the first empty position at the end of the queue.
- ► A 2-packet is always stored at the end of the queue in the usual FIFO manner.
- ► If the server becomes available, the first packet in the queue (either type 1 or type 2) if any is present, is chosen for service during the next slot.

An important parameter is N, the number of reservations in the queue, which is initially chosen and stays fixed throughout the operation of the system. The analysis in Sec. 4.1 only considered the model for $N = 1$, but was substantially extended in Sec. 4.3 to be able to account for *multiple* reserved spaces ($N > 1$). For the packets of type 1, we have obtained a recursive relation for the pgf of their delay distribution while a closed-form expression was derived for the delay distribution of the 2-packets. For both types, we have provided algorithms to calculate the mean value and the approximated tail distribution of the delay.

Furthermore, we have proven that the queue under the Reservation discipline decouples into two logical sub-queues when N is very large. One of these sub-queues contains only 1-packets, while the other contains the 2-packets. These logical sub-queues behave exactly the same as the high-priority and low-priority queues under the Absolute Priority discipline (AP). We have shown that the convergence towards AP-behaviour as a function of N is *slower* when both the total load λ_T and/or the traffic mix λ_1/λ_T is larger.

Some of the advantages of the Reservation discipline as described here are the following. First, it is easy to implement, unlike many other priority scheduling disciplines like Weighted Fair Queueing (WFQ) or its variants. Secondly, we have demonstrated that the number of reservations in the system can be seen as a *control parameter* by which the delay differentiation can be fine-tuned. This feature allows for dynamic queue management in case the delay requirements of the two types change. Finally, the decay rate of the tail distribution of both type 1 and type 2 delay is always the same as under FIFO. Specifically, this means that in case the arrival distribution is well-behaved, the tail of the delay distribution exhibits an exponential decay, which is not necessarily the case under AP.

Bibliography

Author's list of publications

The results in this thesis have all been published in the form of contributions to international scientific journals and conferences, with the exception of Secs. 4.2 and 4.3. Parts of this work have also been used in the framework of the European projects COST-257 and its successor COST-279. The study of Stop-and-Wait ARQ was part of our contribution to the IAP 'MOTION' project supported by the Belgian Science Policy.

- With regard to the correlated train arrival model from chapter 2, the mean values of respectively the number of *packets* in the system u and their delay d are calculated in [18]. In [5, 17, 19] we also determine the variance and the approximated tail distribution of u. The mean values of the delay c and the transmission time h experienced by an arbitrary *message* are found in [15, 16] and in [4]. In the latter paper, we also present the numerical algorithm for calculating the approximated tail distribution of c.

- The queueing analysis of the Stop-and-Wait ARQ protocol from chapter 3 is published in [9, 11] with as results the pgf, the mean value and the approximated tail distribution of the number of packets u in the transmitter queue. In [8], we concentrate specifically on the calculation of the throughput η of the protocol, as well as the pgf and the tail distribution of the packet delay d. In [1], all of the above results are brought together. Additionally, [1] contains an analysis of the queue content in case the errors in the channel are strongly correlated. The calculations in [1, 8] are presented in a more concise and tractable way than in [9, 11] because of the introduction of more efficient notation.

- The analysis of the reservation queue as presented in chapter 4 in case of only a single reserved space in the queue is published in [6, 10], where we obtain the pgf, the mean value, the variance and the approximated tail distribution of the delay experienced by either type of packets. An abridged overview of the analysis and its results are also given in [2].

- The publications [3, 7, 12] and [13] are joint work with dr. D. Fiems and prof. R. Boel respectively, of which the subject is not within the scope of this thesis.

Journal publications

[1] S. De Vuyst, S. Wittevrongel, H. Bruneel, *Performance analysis of the stop-and-wait ARQ protocol over a channel with bursty errors*, HET-NETs 2004 special journal issue, extended version, accepted.

[2] S. De Vuyst, S. Wittevrongel, H. Bruneel, *Delay differentiation by reserving a space in the queue*, Electronics Letters, 2005, Vol. 41, No. 9, pp. 69–70.

[3] D. Fiems, S. De Vuyst, H. Bruneel, *The combined gated-exhaustive vacation system in discrete-time*, Performance Evaluation, 2002, Vol. 49, No. 1–4, pp. 227–239.

[4] S. De Vuyst, S. Wittevrongel, H. Bruneel, *Mean value and tail distribution of the message delay in statistical multiplexers with correlated train arrivals*, Performance Evaluation, 2002, Vol. 48, No. 1–4, pp. 103-129.

[5] S. De Vuyst, S. Wittevrongel, H. Bruneel, *Statistical multiplexing of correlated variable-length packet trains: an analytic performance study*, Journal of the Operational Research Society (JORS), 2001, Vol. 52, No. 3, pp. 318–327.

Conference contributions

[6] S. De Vuyst, S. Wittevrongel, H. Bruneel, *Analysis of a priority scheduling discipline with place reservation*, Proceedings of INOC 2005, International Network Optimization Conference (20–23 March 2005, Lisbon, Portugal), pp. 442–448.

[7] D. Fiems, S. De Vuyst, H. Bruneel, *Performance analysis of a video streaming buffer*, Proceedings of the 4th International Conference on Networking, Part I (17–21 April 2005, Reunion Island), pp. 892–900.

[8] S. De Vuyst, S. Wittevrongel, H. Bruneel, *Delay analysis of the Stop-and-Wait ARQ Protocol over a correlated error channel*, Proceedings of HET-NETs 2004, Performance Modelling and Evaluation of Heterogeneous Networks (26–28 July 2004, Ilkley, West Yorkshire, UK), pp. 21/1–21/11.

[9] K. Tworus, S. De Vuyst, S. Wittevrongel, H. Bruneel, *Transmitter buffer behavior of the stop-and-wait ARQ scheme under correlated errors*, Proceedings of DASD 2004, Conference on Design, Analysis, and Simulation of Distributed Systems (18–22 April 2004, Arlington, Washington D.C., USA), pp. 10–18.

[10] S. De Vuyst, S. Wittevrongel, H. Bruneel, *A queueing discipline with place reservation*, Proceedings of the COST 279 11th Management Committee Meeting (23–24 September 2004, Gent, Belgium), COST279TD(04)36.

[11] K. Tworus, S. De Vuyst, S. Wittevrongel, H. Bruneel, *Queueing analysis of the stop-and-wait ARQ protocol in a wireless environment*, Proceedings of the COST 279 8th Management Committee Meeting (25–26 September 2003, Warsaw, Poland), COST279TD(03)43.

[12] D. Fiems, S. De Vuyst, H. Bruneel, *Discrete-time analysis of the gated-exhaustive vacation queue*, Proceedings of the COST 279 Sixth Management Committee Meeting (22–23 January 2003, Dubrovnik, Croatia), COST279TD(03)17.

[13] R. Boel, S. De Vuyst, *Prediction based resource allocation, a simulation experiment*, Proceedings of the COST 279 3rd Management Committee Meeting (7–8 February 2002, Leidschendam, The Netherlands), COST279TD(02)08.

[14] S. De Vuyst, S. Wittevrongel, H. Bruneel, *Queueing of correlated packet-trains*, Abstracts 2nd FTW PhD Symposium (12 December 2001, Gent University, Belgium), p. 32.

[15] S. De Vuyst, S. Wittevrongel, H. Bruneel, *Message delays and transmission times in statistical multiplexers with correlated train arrivals*, Proceedings of the 8th IFIP Workshop on Performance Modelling and Evaluation of ATM & IP Networks, IFIP ATM&IP 2000 (17–19 July 2000, Ilkley, West Yorkshire, UK), pp. 87/1–87/13.

[16] S. De Vuyst, S. Wittevrongel, H. Bruneel, *Mean delay and transmission time of correlated variable-length messages in a discrete-time queue*, Proceedings of the COST 257 12th Management Committee Meeting (18–19 May 2000, Kjeller, Norway), COST257TD(00)33.

[17] S. De Vuyst, S. Wittevrongel, H. Bruneel, *Queueing analysis of discrete-time buffer systems with correlated packet-train arrivals*, Abstract Booklet of the OR41 Conference (14–16 September 1999, Edinburgh, Scotland, UK), p. 78.

[18] S. Wittevrongel, S. De Vuyst, H. Bruneel, *Mean buffer contents and mean packet delay for statistical multiplexers with correlated train arrivals*, Proceedings of the 7th IFIP Workshop on Performance Modelling and Evaluation of ATM & IP Networks, IFIP ATM&IP ’99 (28–30 June 1999, Antwerp, Belgium), pp. 1/12–12/12.

[19] S. De Vuyst, S. Wittevrongel, H. Bruneel, *Moments and tail distribution of the buffer contents in statistical multiplexers with correlated variable-length messages*, Proceedings of the COST 257 10th Management Committee Meeting (30 Sept. – 1 Oct. 1999, Larnaca, Cyprus), COST257TD(99)40.

References

[20] J. Abate, W. Whitt, *Numerical Inversion of Probability Generating Functions*, Operations Research Letters, Vol. 12, No. 4, 1992, pp. 245–251.

[21] D. Anick, D. Mitra, M.M. Sondhi, *Stochastic Theory of a Data Handling System with Multiple Sources*, Bell Systems Technical Journal, 1982, Vol. 61, pp. 1871–1894.

[22] M.E. Anagnostou, E.N. Protonotarios, *Performance Analysis of the Selective Repeat ARQ Protocol*, IEEE Transactions on Communications, Vol. 34, No. 2, 1986, pp. 127–135.

[23] A. Annamalai, V.K. Bhargava, *Analysis and Optimization of Adaptive Multicopy Transmission ARQ Protocols for Time-Varying Channels*, IEEE Transactions on Communications, Vol. 46, No. 10, 1998, pp. 1356–1368.

[24] A. Annamalai, V.K. Bhargava, W.S. Lu, *On Adaptive Go-Back- ARQ Protocol for Variable-Error Rate Channels*, IEEE Transactions on Communications, Vol. 46, No. 11, 1998, pp. 1405–1408.

[25] J. Aráuz, P. Krishnamurthy, *Markov Modeling of 802.11 Channels*, Proceedings of the IEEE Vehicular Technology Conference, VTC 2003 (4–9 October 2003, Orlando, Florida, USA).

[26] F. Babich, G. Lombardi, *On Verifying a First-order Markovian Model for the Multi-threshold Success/Failure Process for Rayleigh Channel*, Proceedings of the IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC '97 (1–4 September 1997, Helsinki, Finland).

[27] J.J. Bae, T. Suda, *Survey of Traffic Control Schemes and Protocols in ATM Networks*, Proceedings of the IEEE, Vol. 79, No. 2, 1991, pp. 170–189.

[28] R.J. Benice, A.H. Frey, *An Analysis of Retransmission Systems*, IEEE Transactions on Communication Technology, Vol. CS-12, 1964, pp. 135–145.

[29] R.J. Benice, A.H. Frey, *Comparisons of Error Control Techniques*, IEEE Transactions on Communication Technology, Vol. 12, 1964, pp. 146–154.

[30] A.T. Bharucha-Reid, *Elements of the Theory of Markov Processes and their Applications* (McGraw-Hill, New York, 1960) republished: (Dover, 1997).

[31] N.D. Birell, *Pre-emptive Retransmission for Communication over Noisy Channels*, IEE Proceedings, Part F, Vol. 128, 1981, pp. 393–400.

[32] H. Bischl, E. Lutz, *Packet Error Rate in the Noninterleaved Rayleigh Channel*, *IEEE Transactions on Communications*, Vol. 43, No. 2–4, 1995, pp. 1375–1382.

[33] S. Blake et al., *An Architecture for Differentiated Services*, Internet RFC 2475, December 1998.

[34] P.P. Bocharov, C. D'Apice, A.V. Pechinkin, S. Salerno, *Queueing Theory* (VSP, Leiden, The Netherlands, 2004).

[35] G. Bolch, S. Greiner, H. Meer, K. Trivedi, *Queueing Networks and Markov Chains* (Wiley & Sons, New York, 1998).

[36] F. Brackx, *Wiskundige Analyse III*, Ghent University course notes, 1999.

[37] R. Braden, D. Clark, S. Shenker, *Integrated Services in the Internet Architecture: an Overview*, Internet RFC 1633, June 1994.

[38] E. Brockmeyer, H.L. Halstrøm, A. Jensen, *The Life and Work of A.K. Erlang*, Transactions of the Danish Academy of Technical Sciences, No. 2, 1948.

[39] H. Bruneel, M. Moeneclaey, *On the Throughput Performance of Some Continuous ARQ Strategies with Repeated Transmissions*, IEEE Transactions on Communications, Vol. 34, No. 3, 1986, pp. 244–249.

[40] H. Bruneel, *Transient Queueing Behavior of Buffers with Unreliable Output Line*, Proceedings of ICC '88, the IEEE International Conference on Communications (12–15

June 1988, Philadelphia, USA), pp. 1291–1295.

[41] H. Bruneel, *Throughput Comparison for Stop-and-Wait ARQ Schemes with Memory*, Electronics Letters, Vol. 24, No. 9, 1988, pp. 531–533.

[42] H. Bruneel, J. De Vriendt, C. Ysebaert, *Receiver Buffer Behavior for the Selective-Repeat ARQ Protocol*, Computer Networks and ISDN Systems, Vol. 19, No. 2, 1990, pp. 129–142.

[43] H. Bruneel, B.G. Kim, *Discrete-Time Models for Communication Systems Including ATM* (Kluwer Academic Publishers, Boston, 1993).

[44] H. Bruneel, *Performance of Discrete-Time Queueing Systems*, Computers & Operations Research, Vol. 20, No. 3, 1993, pp. 303–320.

[45] H. Bruneel, *Packet Delay and Queue Length for Statistical Multiplexers with Low-speed Access Lines*, Computer Networks and ISDN Systems, Vol. 25, No. 12, 1993, pp. 1267–1277.

[46] H. Bruneel, B. Steyaert, E. Desmet, G. Petit, *Analytic Derivation of Tail Probabilities for Queue Lengths and Waiting Times in ATM Multiserver Queues*, European Journal of Operational Research, Vol. 76, 1994, pp. 563-572.

[47] H. Bruneel, *Calculation of Message Delays and Message Waiting Times in Switching Elements with Slow Access Lines*, IEEE Transactions on Communications, Vol. COM-42, No. 2–4, 1994, pp. 255–259.

[48] H. Bruneel, C. Tison, *Improving the Throughput of Stop-and-Wait ARQ Schemes with Repeated Transmissions*, AEÜ International Journal of Electronics and Communications, Vol. 51, No. 1, 1997, pp. 1–8.

[49] W. Burakowski, H. Tarasiuk, *On New Strategy for Prioritising the Selected Flow in Queuing System*, Proceedings of the COST 257 11th Management Committee Meeting, (20–21 January 2000, Barcelona, Spain), COST-257 TD(00)03.

[50] W. Burakowski, M. Fudala, *Priority Forcing Scheme: a New Strategy for Getting Better than Best Effort Service in IP-Based Network*, Proceedings of the IFIP Workshop on Internet Technologies WITASI 2002, (10-11 October 2002, Wroclaw, Poland), pp. 135–150.

[51] Y. Chang, C. Leung, *On Weldon's ARQ Strategy*, IEEE Transactions on Communications, Vol. 32, No. 3, 1984, pp. 297–300.

[52] A.M. Chen, R.R. Rao, *On Tractable Wireless Channel Models*, Proceedings of the 9th IEEE Internatonal Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC98, (8–11 September 1998, Boston, MA, USA).

[53] Y.J. Cho, C.K. Un, *Performance Analysis of ARQ Error Controls under Markovian Block Error Pattern*, IEEE Transactions on Communications, Vol. 42, 1994, pp. 2051–2061.

[54] B.D. Choi, D.I. Choi, Y. Lee, D.K. Sung, *Priority Queueing System with Fixed-length Packet-train Arrivals*, IEE Proceedings-Communications, Vol. 145, No. 5, 1998, pp. 331–336.

[55] L. Chuang, L. Wanming, Y. Baoping, S. Chanson, *A Dynamic Partial Buffer Sharing Scheme for Packet Loss Control in Congested Networks*, Proceedings of the International Conference on Communication Technology, ICCT 2000 (21–25 August 2000, Beijing, China), Vol. 2, pp. 1286–1293.

[56] W.W. Chu, A.G. Konheim, *On the Analysis and Modeling of a Class of Computer Communication Systems*, IEEE Transactions on Communications, 1972, Vol. 20, No. 3, pp. 645–660.

[57] I. Cidon, A. Khamisy, M. Sidi, *Dispersed Messages in Discrete-time Queues: Delay, Jitter and Threshold Crossing*, Proceedings of INFOCOM '94 (12–16 June 1994, Toronto, Canada), pp. 218–223.

[58] R.H. Clarke, *A Statistical Theory of Mobile Radio Reception*, Bell Systems Technical

Journal, Vol. 47, No. 7, 1968, pp. 957–1000.

[59] R.B. Cooper, *Introduction to Queueing Theory, 2nd Edition* (North Holland, New York, 1981).

[60] E. Costamagna, L. Favalli, P. Gamba, P. Savazzi, *Block-Error Probabilities for Mobile Radio Channels Derived from Chaos Equations*, *IEEE Communications Letters*, Vol. 3, No. 3, March 1999, pp. 66–68.

[61] D.J. Costello Jr., J. Hagenauer, H. Imai, S.B. Wicker, *Applications of Error-Control Coding*, *IEEE Transactions on Information Theory*, Vol. 44, No. 6, 1998, pp. 2531–2560.

[62] D. Cox, *The Analysis of non-Markovian Stochastic Processes by the Inclusion of Supplementary Variables*, *Proceedings of the Cambridge Philosophical Society*, Vol. 51, 1955, pp. 433–441.

[63] J.N. Daigle, *Message Delays at Packet-switching Nodes Serving Multiple Classes*, *IEEE Transactions on Communications*, Vol. 38, No. 4, 1990, pp. 447–455.

[64] J.D. Daigle, *Queueing Theory With Applications To Packet Telecommunication*, (Springer Verlag, 2005).

[65] R. Dalke, G. Hufford, *Analysis of the Markov Character of a General Rayleigh Fading Channel*, *NTIA Technical Memorandum*, TM-05-423, April 2005.

[66] T. Daniëls, C. Blondia, *Asymptotic Behavior of a Discrete-time Queue with Long Range Dependent $M|G|\infty$ Input*, *Proceedings of the COST-257 Seventh Management Committee Meeting* (24–25 September 1998, Granada, Spain), COST257TD(98)50.

[67] T. Daniëls, C. Blondia, *Asymptotic Behavior of a Discrete-time Queue with Long Range Dependent Input*, Proceedings of INFOCOM '99 (21–25 March 1999, New York, USA), pp. 633–640.

[68] T. Daniels, *Asymptotic Behaviour of Queueing Systems*, PhD thesis, University of Antwerp, Belgium, 1999.

[69] A. Demers, S. Keshav, S. Shenker, *Analysis and Simulation of a Fair Queueing Algorithm*, *Proceedings of the ACM Symposium on Communications Architectures & Protocols*, SIGCOMM '89 (19–22 September 1989, Austin, TX, USA), pp. 1–12.

[70] M. De Munnynck, S. Wittevrongel, A. Lootens, H. Bruneel, *Queueing Analysis of Some Continuous ARQ Strategies with Repeated Transmissions*, *IEE Electronics Letters*, Vol. 38, No. 21, 2002, pp. 1295–1297.

[71] M. De Munnynck, S. Wittevrongel, A. Lootens, H. Bruneel, *Transmitter Buffer Behaviour of Stop-and-Wait ARQ Schemes with Repeated Transmissions*, *IEE Proceedings - Communications*, vol. 149, no. 1, 2002, pp. 13–17.

[72] P. Dent, G.E. Bottomley, T. Croft, *Jakes Fading Model Revisited*, *Electronics Letters*, Vol. 29, No. 13, June 1993, pp. 422–431.

[73] C. Dovrolis, D. Stiliadis, P. Ramanathan, *Proportional Differentiated Services: Delay Differentiation and Packet Scheduling*, *ACM Computer Communications Review*, Vol. 29, No. 4, 1999, pp. 109–120.

[74] M. Drmota M., *A Bivariate Asymptotic Expansion of Coefficients of Powers of Generating Functions*, *European Journal of Combinatorics*, Vol. 15, 1994, pp. 139–152.

[75] R. Dube, C.D. Rais, S.K. Tripathi, *Improving NFS Performance over Wireless Links*, *IEEE Transactions on Computing*, Vol. 46, No. 3, 1997, pp. 290–298.

[76] E.O. Elliott, *Estimates of Error Rates for Codes on Burst-Noise Channels*, *Bell Systems Technical Journal*, Vol. 42, No. 9, 1963, pp. 1977–1997.

[77] Y. Ephraim, N. Merhav, *Hidden Markov Processes*, *IEEE Transactions on Information Theory*, Vol. 48, No. 6, 2002, pp. 1518–1569.

[78] A.K. Erlang, *The Theory of Probabilities and Telephone Conversations*, *Nyt Tidsskrift for Matematik B*, Vol. 20, 1909.

[79] R. Fantacci, *Performance Evaluation of Some Efficient Stop-and-Wait Techniques*, IEEE Transactions on Communications, Vol. 40, No. 11, 1992, pp. 1665–1669.

[80] R. Fantacci, *Queueing Analysis of the Selective Repeat Automatic Repeat Request Protocol in Wireless Packet Networks*, IEEE Transactions on Vehicular Technology, Vol. 45, No. 2, 1996, pp. 258–264.

[81] G. Fayolle, E. Gelenbe, G. Pujolle, *An Analytic Evaluation of the Performance of the "Send and Wait" Protocol*, IEEE Transactions on Communications, Vol. 26, No. 3, 1978, pp. 313–319.

[82] W. Feller, *An Introduction to Probability Theory and its Applications, Vol. I*, 3rd edition (Wiley & Sons, New York, 1968).

[83] D. Fiems, H. Bruneel, *A Note on the Discretization of Little's Result*, Operations Research Letters, Vol. 30, No. 1, 2002, pp. 17–18.

[84] P. Flajolet, A.M. Odlyzko, *Singularity Analysis of Generating Functions*, SIAM Journal for Discrete Mathematics, Vol. 3, No. 2, pp. 216–240.

[85] P. Flajolet, R. Sedgewick, *Analytic Combinatorics* – book in preparation (http://pauillac.inria.fr/algo/flajolet).

[86] F. J. Flanigan, *Complex Variables: Harmonic and Analytic Functions* (Dover, New York, 1983).

[87] S. Floyd, V. Jacobson, *Random Early Detection Gateways for Congestion Avoidance*, IEEE-ACM Transactions on Networking, Vol. 1, No. 4, August 1993, pp. 397–413.

[88] F.G. Foster, *On the Stochastic Matrices Associated with Certain Queuing Processes*, Annals of Mathematical Statistics, 1953, Vol. 25, pp. 355–360.

[89] B.D. Fritchman, *A Binary Channel Characterization Using Partitoned Markov Chains*, IEEE Transactions on Information Theory, Vol. 13, No. 2, 1967, pp. 221–227.

[90] C. Fujiwara, M. Kasahara, K. Yamashita, T. Namekawa T., *Evaluations of Error Control Techniques in Both Independent-Error and Dependent-Error Channels*, IEEE Transactions on Communications, Vol. 26, No. 6, 1978, pp. 785–793.

[91] H.R. Gail, S.L. Hantler, B.A. Taylor, *On a Preemptive Markovian Queue with multiple servers and two priority classes*, Mathematics of Operations Research, 1992, Vol. 17,No. 2, pp. 365–391.

[92] H.R. Gail, S.L. Hantler, B.A. Taylor, *Spectral Analysis of M/G/1 and G/M/1 type Markov Chains*, Advances in Applied Probability, 1996, Vol. 28, pp. 114–165.

[93] R.G. Gallager, *Information Theory and Reliable Communication* (Wiley & Sons, New York, 1968).

[94] F.R. Gantmacher, *The Theory of Matrices, Volume One* (AMS Chelsea Publishing, Providence, Rhode Island, 1959).

[95] J. Garcia-Frias, PM. Crespo, *Hidden Markov Models for Burst Error Characterization in Indoor Radio Channels*, IEEE Transactions on Vehicular Technology, Vol. 46, No. 4, 1997, pp. 1006–1020.

[96] P. Gevros, J. Crowcroft, P. Kirstein, S. Bhatti, *Congestion Control Mechanisms and the Best Effort Service Model*, IEEE Network, May-June 2001, pp. 16–26.

[97] E.N. Gilbert, *Capacity of a Burst-noise Channel*, Bell Systems Tech. Journal, Vol. 39, No. 9, 1960, pp. 1253–1265.

[98] A.J. Goldsmith, P.P. Varaiya, *Capacity, Mutual Information, and Coding for Finite-State Markov Channels*, IEEE Transactions on Information Theory, Vol. 42, No. 5, 1996, pp. 868–886.

[99] M. O. González, *Classical Complex Analysis*, (Marcel Dekker, New York, 1992).

[100] R.M. Gray, *Probability, Random Processes, and Ergodic Properties* (Springer Verlag, 1988).

[101] M.O. Hasna, M.-S. Alouini, *Application of the Harmonic Mean Statistics to the End-to-*

end Performance of Transmission Systems with Relays, Proceedings of the IEEE Global Telecommunications Conference (17–21 November 2002, Taipei, Taiwan), pp. 1310-1314.

[102] J.F. Hayes, *Modeling and Analysis of Computer Communications Networks* (Plenum Press, New York, 1984).

[103] G.J. Heijenk, M. El Zarki, I.G. Niemegeers, *Modelling of Segmentation and Reassembly Processes in Communication Networks*, Proceedings of the 14th International Teletraffic Congress, ITC 14 (6–10 June 1994, Antibes Juan-les-Pins, France), Elsevier, Teletraffic Science and Engineering, pp. 513–524.

[104] M. Hueda, *On the Markovian Approximation for Block-errors in DS-CDMA Transmissions over Slow Fading Channels with Multicarrier Transmit Diversity*, Proceedings of the IEEE International Conference on Communications, ICC'02 (28 April 28 – 2 May 2, 2002, New York, USA), Vol. 2. pp. 737–741.

[105] J. Hunter, *Mathematical Techniques of Applied Probability, Volumes 1 & 2*, Operations Research and Industrial Engineering (Academic Press, New York, 1983).

[106] H. Inai, J. Yamakita, *A Two-Layer Queueing Model to Predict Performance of Packet Transfer in Broadband Networks*, Annals of Operations Research, Vol. 79, No. 1, 1998, pp. 349–371.

[107] W.C. Jakes, *Microwave Mobile Communications* (Wiley & Sons, New York, 1974).

[108] H. Jianhua, K.R. Subramanian, D. Donghua, W. Wei, *Novel Methods for the Performance Analysis of Adaptive Hybrid Selective Repeat ARQ*, Computer Communications, Vol. 23, 2000, pp. 1548–1557.

[109] D.G. Kendall, *Stochastic Processes Occurring in the Theory of Queues and their Analysis by Means of the Imbedded Markov Chain*, Annals of Mathematical Statistics, Vol. 24, 1954, pp. 338–354.

[110] J.G. Kim, M. Krunz, *Delay Analysis of Selective Repeat ARQ for a Markovian Source over a Wireless Channel*, Proceedings of the 2nd ACM International Workshop on Wireless Mobile Multimedia WoWMoM (20 August 1999, Seattle, USA), pp. 59–66.

[111] N.K. Kim, M.L. Chaudhry, *Numerical Inversion of Generating Functions – A Computational Experience*, Operations Research – to appear.

[112] Y.Y. Kim, S.Q. Li, *Capturing Important Statistics of a Fading/Shadowing Channel for Network Performance Analysis*, IEEE Journal on Selected Areas in Communications, Vol. 17, No. 5, May 1999, pp. 888–901.

[113] S.R. Kim, C.K. Un, *Throughput Analysis for Two ARQ Schemes Using Combined Transition Matrix*, IEEE Transactions on Communications, Vol. 40, No. 11, 1992, pp. 1679–1683.

[114] L. Kleinrock, *Queueing Systems Volume 1: Theory* (Wiley & Sons, New York, 1975).

[115] V. Klimenok, *On the Modification of Rouche's Theorem for the Queueing Theory Problems*, Queueing Systems, 2001, Vol. 38, No. 4, pp. 431–434.

[116] D.E. Knuth, *The Art of Computer Programming, Vol. 1 – Fundamental Algorithms*, 3rd Ed. (Addison-Wesley, Reading, Massachusetts, 1997).

[117] A.N. Kolmogorov, *Grundbegriffe der Wahrscheinlichkeitsrechnung* (Springer Verlag, Berlin, 1933) English translation: *Foundations of the Theory of Probability, 2nd Ed.* (Chelsea Publishing Company, New York, 1956).

[118] A.G. Konheim A.G., *A Queueing Analysis of Two ARQ Protocols*, IEEE Transactions on Communications, Vol. 28, No. 7, 1980, pp. 1004–1014.

[119] A. Konrad A., B.Y. Zhao, A.D. Joseph, R. Ludwig R., *A Markov-Based Channel Model Algorithm for Wireless Networks*, ACM Wireless Networks, Selected papers from MSWiM 2001, Vol. 9, No. 3, 2003, pp. 189–199.

[120] L. Kosten, *Stochastic Theory of Service Systems* (Pergamon, Oxford, 1973).

[121] L. Kosten, *Stochastic Theory of a Multi-entry Buffer (1)*, Delft Progress Report, 1974, Vol. 1, pp. 10–18.

[122] K. Laevens, H. Bruneel, *Delay Analysis for ATM Queues with Random Order of Service*, Electronics Letters, Vol. 31, 1995, pp. 346–347.

[123] K. Laevens, *Stochastische Modellering van ATM-schakelelementen met Buffers aan de Ingangszijde*, PhD thesis, Ghent University, Belgium, 1999.

[124] G. Latouche, V. Ramaswami, *Introduction to Matrix Geometric Methods in Stochastic Modeling*, ASA-SIAM Series on Statistics and Applied Probability (SIAM, Philadelphia, 1999).

[125] P.H. Leslie, *On the Use of Matrices in Certain Population Mathematics*, Biometrika, 1945, Vol. 33, pp. 183–212.

[126] C.H.C. Leung, Y. Kikumoto, A. Sorensen, *The Throughput Efficiency of the Go-Back-N ARQ Scheme Under Markov and Related Error Structures*, IEEE Transactions on Communications, Vol. 36, No. 2, 1988, pp. 231–234.

[127] L. Li, A.J. Goldsmith, *A Decision-Feedback Maximum-likelihood Decoder for Fading Channels*, Proceedings of the IEEE Global Communication Conference, GLOBECOM '97 (3–8 November 1997, Phoenix, AZ, USA), pp. 332–336.

[128] S.Q. Li, *Generating Function Approach for Discrete Queueing Analysis with Decomposable Arrival and Service Markov chains*, Stochastic Models, 1993, Vol. 9, No. 3, pp. 401–420.

[129] S.Q. Li, H.D. Sheng, *Discrete Queueing Analysis of Multimedia Traffic with Diversity of Correlation and Burstiness Properties*, IEEE Transactions on Communications, 1994, Vol. 42, No. 2/3/4, pp. 1339–1351.

[130] S. Lin, P.S. Yu, *An Effective Error Control Scheme for Satellite Communications*, IEEE Transactions on Communications, Vol. 28, No. 3, 1980, pp. 395–401.

[131] S. Lin, P.S. Yu, *A Hybrid ARQ Scheme with Parity Retransmissions*, IEEE Transactions on Communications, Vol. 30, No. 7, 1982, pp. 1702–1719.

[132] S. Lin, D.J. Costello, M.J. Miller, *Automatic-repeat-request Error-control Schemes*, IEEE Communications Magazine, Vol. 22, No. 12, 1984, pp. 5–17.

[133] D.V. Lindley, *The Theory of Queues with a Single Server*, Proceedings of the Cambridge Philosophical Society, 1952, Vol. 48, No. 2, pp. 277–289.

[134] K.Y. Liu, D.W. Petr, V.S. Frost, H.B. Zhu, C. Braun, W.L. Edwards, *Design and Analysis of a Bandwidth Management Framework for ATM-Based Broadband ISDN*, IEEE Communications Magazine, Vol. 35, No. 5, 1997, pp. 138–145.

[135] D.L. Lu, J.F. Chang, *Analysis of ARQ Protocols via Signal Flow Graphs*, IEEE Transactions on Communications, Vol. 37, No. 3, 1989, pp. 245–251.

[136] D.L. Lu, J.F. Chang, "*Performance of ARQ Protocols in Nonindependent Channel Errors*". IEEE Transactions on Communications, Vol. 41, No. 5, May 1993.

[137] R.H. McCullough, *The Binary Regenerative Channel*, Bell System Technical Journal, Vol. 47, October 1968, pp. 1713–1735.

[138] J. McDougall, S. Miller, *Sensitivity of Wireless Network Simulations to a Two-State Markov Model Channel Approximation*, Proceedings of the IEEE Global Telecommunications Conference, GLOBECOM 2003, (1–5 December 2003, San Francisco, USA).

[139] T. Meisling, *Discrete-time Queueing Theory*, Journal of the Operational Research Society, 1958, Vol. 6, pp. 96–105.

[140] M. Menth, M. Schmid, H. Heiss, T. Reim, *MEDF – A Simple Scheduling Algorithm for Two Real-time Transport Service Classes with Application in the UTRAN*, Proceedings of INFOCOM '03 (30 March – 3 April 2003, San Francisco, USA).

[141] C.D. Meyer, *Matrix Analysis and Applied Linear Algebra*, (SIAM, Philadelphia, 2000).

[142] H. Michiel, K. Laevens, *Teletraffic Engineering in a Broad-band Era*, Proceedings of

the IEEE, 1997, Vol. 85, No. 12, pp. 2007–2032.

[143] P. Van Mieghem, B. Steyaert, G.H. Petit, *Performance of Cell Loss Priority Management Schemes in a Single Server Queue*, International Journal of Communication Systems, Vol. 10, 1997, pp. 161-180.

[144] D. Minoli, E. Minoli, *Delivering Voice over IP Networks* (Wiley & Sons, 1998).

[145] M. Moeneclaey, H. Bruneel, *Efficient ARQ Scheme for High Error Rate Channels*, Electronics Letters, Vol. 20, No. 23, 1984, pp. 986–987.

[146] M. Moeneclaey, H. Bruneel, I. Bruyland and D.Y. Chung, *Throughput Optimization for a Generalized Stop-and-Wait ARQ Scheme*, IEEE Transactions on Communications, Vol. 34, No. 2, February 1986, pp. 205–207.

[147] J.M. Moris, *On Another Go-Back-N ARQ Technique for High Error Rate Conditions*, IEEE Transactions on Communications, Vol. 26, No. 1, 1978, pp. 187–189.

[148] J.M. Moris, *Optimal Blocklengths for ARQ Error Control Schemes*, IEEE Transactions on Communications,Vol. 27, No. 2, 1979, pp. 488–493.

[149] M. Mowbray, G. Karlsson, T. Köhler, *Capacity Reservation for Multimedia Traffic*, Distrib. Syst. Engng., Vol. 5, 1998, pp. 12–18.

[150] R. Mukhtar, M. Zukerman, F. Cameron, *Packet Latency for Type-II Hybrid ARQ Transmissions over a Correlated Error Channel*. Proceedings of European Wireless 2002 Conference (25–28 February 2002, Florence, Italy), pp. 107-113.

[151] M. De Munnynck, A. Lootens, S. Wittevrongel, H. Bruneel *Transmitter Buffer Behaviour of Stop-and-Wait ARQ Schemes with Repeated Transmissions*, IEE Proceedings – Communications, Vol. 149, No. 1, 2002, pp. 13–17.

[152] M. Neuts, *Matrix-Geometric Solutions in Stochastic Models, an Algorithmic Approach* (The Johns Hopkins University Press, 1981).

[153] G.D. Nguyen, *Error Detection Codes: Algorithms and Fast Implementation*, IEEE Transactions on Computers, Vol. 54, No. 1, 2005, pp. 1–11.

[154] K. Nguyen, R.H. Katz, B. Noble, M. Satyanarayanan, *A Trace-based Approach for Modeling Wireless Channel behavior*, Proceedings of the 28th conference on Winter Simulation (8–11 December 1996, Coronado, CA, USA).

[155] M.A. Nielsen, I.L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, 2000).

[156] N. Nikaein, C. Bonnet, *Performance Evaluation of an Adaptive Error Control Protocol in Wireless Networks*, International Symposium on Performance Evaluation of Computer and Telecommunication Systems, SPECTS '01 (15–18 July 2001, Orlando, Florida, USA).

[157] A.M. Odlyzko, *Asymptotic Enumeration Methods*, in Handbook of Combinatorics Vol. 2, Eds. R.L. Graham, M. Groetschel, L. Lovasz (Elsevier, 1995) pp. 1063-1229.

[158] A.G. Pakes, *Some Conditions for Ergodicity and Recurrence of Markov Chains*, Operations Research, 1969, Vol. 17, pp. 1058–1061.

[159] A.K. Parekh, R.G. Gallager, *A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Single Node Case*, IEEE Network, Vol. 1, No. 3, June 1993, pp. 344–57.

[160] M. Patzold, F. Laue, *Statistical Properties of Jakes' Fading Channel Simulator*, Proceedings of the 48th IEEE Vehicular Technology Conference, VTC'98 (18–21 May 1998, Ottawa, Ontario, Canada), pp. 712–718.

[161] D.M. Piscitello, A.L. Chapin, *Open Systems Networking: TCP/IP and OSI* (Addison-Wesley, 1994).

[162] M.F. Pop, N.C. Beaulieu, *Limitations of Sum-of-sinusoids Fading Channel Simulators*, IEEE Transactions on Communications, Vol. 49, No. 4, 2001, pp. 699–708.

[163] D. Qiao, S. Choi, K.G. Shin, *Goodput Analysis and Link Adaptation for IEEE 802.11a*

Wireless LANs, IEEE Transactions on Mobile Computing, Vol. 1, No. 4, October-December 2002, pp. 278–292.

[164] T.S. Rappaport, *Wireless Communications* (Prentice Hall, Upper Saddle River NJ, 1996).

[165] S.O. Rice, *Distribution of the Duration of Fades in Radio Transmission: Gaussian Noise Model*, Bell Systems Technical Journal, Vol. 37, No. 3, 1958, pp. 581–635.

[166] K.W. Richardson, *UMTS Overview*, Electronics & Communication Engineering Journal, Vol. 12, No. 3, June 2000, pp. 93–100.

[167] Z. Rosberg, M. Sidi, *Selective-Repeat ARQ: The Joint Distribution of the Transmitter and the Receiver Resequencing Buffer Occupancies*, IEEE Transactions on Communications, Vol. 38, No. 9, 1990, pp. 1430–1438.

[168] M. Rossi, L. Badia, M. Zorzi, *Exact Statistics of ARQ Packet Delivery Delay over Markov Channels with Finite Round-Trip Delay*, Proceedings of IEEE Globecom 2003 (1–5 December 2003, San Francisco), pp. 3356–3360.

[169] M. Rossi M., M. Zorzi, *Analysis and Heuristics for the Characterization of Selective Repeat ARQ Delay Statistics over Wireless Channels*, IEEE Transactions on Vehicular Technology, Vol. 52, No. 5, 2003, pp. 1365–1377.

[170] B.H. Saeki, I. Rubin I., *An Analysis of a TDMA Channel Using Stop-and-Wait, Block, and Select-and-Repeat ARQ Error Control*, IEEE Transactions on Communications, Vol. 30, No. 5, 1982, pp. 1162–1173.

[171] A.R.K. Sastry, *Improving Repeat-request (ARQ) Performance on Satellite Channels under High Error Rate Conditions*, IEEE Transactions on Communications,Vol. 23, No. 4, 1975, pp. 436–439.

[172] C.E. Shannon, *A Mathematical Theory of Communication*, Bell Systems Technical Journal, Vol. 27, 1948, pp. 379–423, 623–656.

[173] P.S. Sindhu, *Retransmission Error Control with Memory*, IEEE Transactions on Communications, Vol. 25, No. 5, 1977, pp. 473–479.

[174] S. Sivaprakasam, K. Shanmugan, *An Equivalent Markov Model for Burst Errors in Digital Channels*, IEEE Transactions on Communications, Vol. 43, No. 2–4, 1995, pp. 1347–1354.

[175] V. Sivaraman, F. Chiussi, *Providing End-to-End Statistical Delay Guarantees with Earliest Deadline First Scheduling and Per-Hop Traffic Shaping*, Proceedings of INFOCOM 2000 (26–30 March 2000, Tel Aviv, Israel).

[176] B. Sklar, *Rayleigh Fading Channels in Mobile Digital Communications Systems, Part I: Characterization*, IEEE Communications Magazine, Vol. 35, No. 7, 1997, pp. 90–100.

[177] S. Stidham, *Analysis, Design, and Control of Queueing Systems*, Operations Research, 2002, Vol. 50, No. 1 (Special 50th Anniversary Issue), pp. 197–216.

[178] A. Striegel, G. Manimaran, *Packet Scheduling with Delay and Loss Differentiation*, Computer Communications, Vol. 25, pp. 21–31.

[179] R.D. Stuart, *An Insert System for Use with Feedback Communication Links*, IEEE Transactions on Communication Systems, Vol. CS-11, 1963, pp. 142–143.

[180] F. Swarts, H.C. Ferreira, *Markov Characterization of Channels with Soft Decision Outputs*, IEEE Transactions on Communications, Vol. 41, No. 5, 1993, pp. 678–682.

[181] H. Takagi, *Queueing Analysis: a Foundation of Performance Evaluation, Volume 1* (North Holland, 1991).

[182] T. Takine, B. Sengupta, T. Hasegawa, *An Analysis of a Discrete-Time Queue for Broadband ISDN with Priorities Among Traffic Classes*, IEEE Transactions on Communications, Vol. 42, No. 2–4, 1994, pp. 1837–1845.

[183] C.C. Tan, N.C. Beaulieu, *On First-Order Markov Modeling for the Rayleigh Fading Channel*, IEEE Transactions on Communications, Vol. 48, No. 12, 2000, pp. 2032–2040.

[184] A.S. Tanenbaum, *Computer Networks, 4th edition* (Prentice Hall, Upper Saddle River NJ, 2003).

[185] C. Tison, H. Bruneel, *Selective Repeat ARQ with Memory*, AEÜ International Journal of Electronics and Communications, Vol. 48, No. 1, 1994, pp. 55–57.

[186] E.C. Titchmarsh, *The Theory of Functions, 2nd edition*, (Oxford University Press, 1939).

[187] D. Towsley, J.K. Wolf, *On the Statistical Analysis of Queue Lengths and Waiting Times for Statistical Multiplexers with ARQ Retransmission Schemes*, IEEE Transactions on Communications, Vol. 27, No. 4, 1979, pp. 693–702.

[188] D. Towsley, *The Analysis of a Statistical Multiplexer with Nonindependent Arrivals and Errors*, IEEE Transactions on Communications, Vol. 28, No. 1, 1980, pp. 65–72.

[189] D. Towsley, *A Statistical Analysis of ARQ Protocols Operating in a Nonindependent Error Environment*, IEEE Transactions on Communications, Vol. 27, No. 7, July 1981, pp. 971–981.

[190] W. Turin, R. Van Nobelen, *Hidden Markov Modeling of Flat Fading Channels*, IEEE Journal on Selected Areas in Communication, Vol. 16, No. 9, 1998, pp. 1809–1817.

[191] W. Turin, *Throughput Analysis of Go-Back-N Protocol in Fading Radio Channels*, IEEE Journal on Selected Areas in Communication, Vol. 17, No. 5, 1999, pp. 881–887.

[192] P.F. Turney, *An Improved Stop-and-Wait ARQ Logic for Data Transmission in Mobile Radio Systems*, IEEE Transactions on Communications, Vol. 29, No. 1, 1981, pp. 68–71.

[193] B. Vinck, H. Bruneel, *Delay Analysis for Single Server Queues*, Electronics Letters, Vol. 32, No. 9, 1996, pp. 802–803.

[194] A.M. Viterbi, *Approximate Analysis of Time-synchronous Packet Networks*, IEEE Journal on Selected Areas in Communications, Vol. 4, No. 6, 1986, pp. 879–890.

[195] V. Vitsas, A.C. Boucouvalas, *Automatic Repeat Request Schemes for Infrared Wireless Communications*, *IEE Electronics Letters*, Vol. 38, No. 5, 2002, pp. 254–246.

[196] J. Walraevens, D. Fiems, H. Bruneel, *Analysis of Queues with a Priority Scheduling Discipline*, Proceedings of the First International Working Conference on Performance Modelling and Evaluation of Heterogeneous Networks HET-NETs '03 (21–23 July 2003, Ilkley, West Yorkshire, UK), Tutorial Papers, pp. T4/1–T4/34.

[197] J. Walraevens, B. Steyaert, H. Bruneel, *Performance Analysis of a Single-Server ATM Queue with a Priority Scheduling*, Computers & Operations Research, Vol. 30, No. 12, 2003, pp. 1807–1829.

[198] J. Walraevens, D. Fiems, H. Bruneel, *Transient Analysis of a Discrete-time Priority Queue*, Proceedings of ASMTA 2005, 12th International Conference on Analytical and Stochastic Modelling Techniques and Applications (1–4 June 2005, Riga, Latvia) accepted.

[199] J. Walraevens, *Discrete-time Queueing Models with Priorities*, PhD thesis, Ghent University, Belgium, 2004.

[200] H.S. Wang, N. Moayeri, *Finite State Markov Channel – A Useful Model for Radio Communication Channels*, IEEE Transactions on Vehicular Technology, Vol. 44, No. 1, 1995, pp. 163–171.

[201] H.S. Wang, P.C. Chang, *On Verifying the First-Order Markovian Assumption for a Rayleigh Fading Channel Model*, IEEE Transactions on Vehicular Technology, Vol. 45, No. 5, 1996, pp. 353–357.

[202] Y.M. Wang, S. Lin, *A Modified Selective-Repeat Type-II Hybrid ARQ System and its Performance Analysis*, IEEE Transactions on Communications, Vol. 31, No. 5, 1983, pp. 593–608.

[203] E.J. Weldon, *An Improved Selective-Repeat ARQ Strategy*, IEEE Transactions on Communications, Vol. 30, No. 3, March 1982.

[204] K. Wesolowski, *Mobile Communication Systems* (Wiley & Sons, New York, 2002).

[205] L. Wilhelmsson, L.B. Milstein, *On the Effect of Imperfect Interleaving for the Gilbert-Elliott Channel*, IEEE Transactions on Communications, Vol. 47, No. 5, 1999, pp. 681–688.

[206] A. Willig, *A New Class of Packet- and Bit-level Models for Wireless Channels*, Proceedings of the 13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC 2002 (15–18 September 2002, Lissabon, Portugal).

[207] S. Wittevrongel, H. Bruneel, *Correlation Effects in ATM Queues Due to Data Format Conversions*, Performance Evaluation, Vol. 32, No. 1, 1998, pp. 35–56.

[208] S. Wittevrongel, *Discrete-time Buffers with Variable-length Train Arrivals*, Electronics Letters, Vol. 34, No. 18, 1998, pp. 1719–1721.

[209] J.M. Wozencraft, M. Horstein, *Digitalized Communication over Two-way Channels*, Fourth London Symposium on Information Theory (September 1960, London, UK).

[210] Y. Xia, D. Tse D., *Analysis on Packet Resequencing for Reliable Network Protocols*, Proceedings of INFOCOM '03 (30 March – 3 April 2003, San Francisco, USA).

[211] Y. Xiong, H. Bruneel, *Buffer Contents and Delay for Statistical Multiplexers with Fixed-length Packet-train Arrivals*, Performance Evaluation, Vol. 17, No. 1, 1993, pp. 31–42.

[212] S.F. Yashkov, *Processor sharing queues: some progress in analysis*, Queueing Systems, 1987, Vol. 2, pp. 1–17.

[213] J.R. Yee, E.J. Weldon, *Evaluation of the Performance of Error-correcting Codes on a Gilbert Channel*, IEEE Transactions on Communications, Vol. 43, No. 8, 1995, pp. 2316–2323.

[214] M. Yoshimoto, T. Takine, Y. Takahashi, T. Hasegawa, *Waiting Time and Queue Length Distributions for Go-Back-N and Selective-Repeat ARQ Protocols*, IEEE Transactions on Communications, Vol. 41, No. 11, 1993, pp. 1687–1693.

[215] P.S. Yu, S. Lin, *An Efficient Selective-Repeat ARQ Scheme for Satellite Channels and Its Throughput Analysis*, IEEE Transactions on Communications, Vol. 29, No. 3, 1981, pp. 353–363.

[216] Q. Zhang, T.F. Wong, J.S. Lehnert, *Performance of a Type-II Hybrid ARQ Protocol in Slotted DS-SSMA Packet Radio Systems*, IEEE Transactions on Communications, Vol. 47, No. 2, 1999, pp. 281–290.

[217] Q. Zhang Q., S.A. Kassam, *Finite-state Markov Model for Rayleigh Fading Channels*, IEEE Transactions on Communications, Vol. 47, No. 11, 1999, pp. 1688–1692.

[218] L. Zhang, *Virtual clocks: a new traffic control algorithm for packet switching networks*, Proceedings of the ACM Symposium on Communications Architectures & Protocols, SIGCOMM '90 (24–27 September 1990, Philadelpia, PA, USA), pp. 19–29.

[219] Z. Zhang, *Queuing Analysis of a Statistical Multiplexer with Multiple Slow Terminals*, Proceedings of ACM SIGCOMM Conference '91 (3–6 September 1991, Zürich, Switzerland), pp. 81–88.

[220] Y.R. Zheng, C. Xiao, *Simulation Models With Correct Statistical Properties for Rayleigh Fading Channels*, IEEE Transactions on Communications, Vol. 51, No. 6, June 2003, pp. 920–928.

[221] H. Zimmermann, *OSI Reference Model – The ISO Model of Architecture for Open Systems Interconnection*, IEEE Transactions on Communications, 1980, Vol. 28, No. 4, pp. 425–432.

[222] M. Zorzi M., R.R. Rao, *Throughput Analysis of Go-Back-N ARQ in Markov Channels with Unreliable Feedback*, Proceedings of IEEE ICC'95 (18–22 June 1995, Seattle, WA, USA), pp. 1232–1237.

[223] M. Zorzi, R.R. Rao, L.B. Milstein, *ARQ Error Control for Fading Mobile Radio Channels*, IEEE Transactions on Vehicular Technology, Vol. 46, No. 5, 1997, pp. 445–455.

[224] M. Zorzi, R.R. Rao, *On Channel Modeling for Delay Analysis of Packet Communications over Wireless Links*, Proceedings of the 36th Allerton Conference on Communications, Control and Computing (23–25 September 1998, Monticello, Illinois, USA).

[225] M. Zorzi, R.R. Rao, *Perspectives on the Impact of Error Statistics on Protocols for Wireless Networks*, IEEE Personal Communications, October 1999, pp. 32–40.